BEA WebLogic Adapter for SAP User's Guide

DN3501342.0406

April 19, 2006

Preface

This document is written for system integrators who develop client interfaces between SAP and other applications. It describes how to use the BEA WebLogic Adapter for SAP to integrate SAP IDocs, RFCs, and BAPIs with your BEA WebLogic Server. It is assumed that readers understand Web technologies and have a general understanding of Microsoft Windows and UNIX systems.

How This Manual Is Organized

The following table lists the titles and numbers of the chapters and the appendix for this manual with a brief description of the contents of each chapter or appendix.

Chapter/Appendix		**Contents**
1	Introducing the BEA WebLogic Adapter for SAP	Provides an overview of the BEA WebLogic Adapter for SAP. Discusses key features and functionality of the adapter.
2	Configuring SAP Inbound Processing	Describes how to configure your SAP system for inbound (client) processing.
3	Creating XML Schemas for SAP	Describes how to create XML schemas for SAP business objects using Servlet Application Explorer.
4	Creating and Publishing Integration Business Services	Describes how to create and publish Integration Business Services using Servlet Application Explorer.
5	Configuring the Event Adapter for SAP	Describes how to create ports and channels to listen for SAP events using Servlet Application Explorer.
6	Using Integration Business Services Policy-Based Security	Describes how to configure Web services policy-based security using Servlet Application Explorer.
7	Management and Monitoring	Describes how to use the management and monitoring tools provided by the Integration Business Services Engine and the JCA Test Tool.
8	Understanding SAP Events	Describes how to configure and test your SAP system for event processing.

Chapter/Appendix		Contents
9	Troubleshooting and Error Messages	Describes limitations and workarounds when connecting to SAP. The adapter-specific errors listed in this chapter can arise whether using the adapter with a JCA, or with an iBSE configuration.
A	Using Application Explorer in BEA WebLogic Workshop to Create XML Schemas and Web Services	Describes how to use Application Explorer in BEA WebLogic Workshop to create XML schemas for SAP BAPIs, RFCs, and IDocs.
B	Using Application Explorer in BEA WebLogic Workshop for Event Handling	Describes how to use Application Explorer in BEA WebLogic Workshop to create events for SAP. In addition, this section provides information on using events in a clustered BEA WebLogic environment.
C	Using WebLogic Workshop to Access Web Services	Describes how to access Web services created for an SAP Business Application Programming Interface (BAPI) and an SAP Remote Function Call (RFC) using BEA WebLogic Workshop.
D	Sample Files and Coding Techniques	Provides sample request and response documents sent between SAP and the BEA WebLogic Adapter for SAP as well as a sample RFC module.

Documentation Conventions

The following table lists and describes the conventions that apply throughout this manual.

Convention	Description
`THIS TYPEFACE` or `this typeface`	Denotes syntax that you must enter exactly as shown.
`this typeface`	Represents a placeholder (or variable) in syntax for a value that you or the system must supply.
`underscore`	Indicates a default setting.
this typeface	Represents a placeholder (or variable), a cross-reference, or an important term.
this typeface	Highlights a file name or command.

Convention	Description
Key + Key	Indicates keys that you must press simultaneously.
`{ }`	Indicates two or three choices; type one of them, not the braces.
`|`	Separates mutually exclusive choices in syntax. Type one of them, not the symbol.
`...`	Indicates that you can enter a parameter multiple times. Type only the parameter, not the ellipsis points (…).
. . .	Indicates that there are (or could be) intervening or additional commands.

Contact Us!

Your feedback on the BEA WebLogic Adapter for SAP documentation is important to us.

Send us e-mail at docsupport@bea.com if you have questions or comments. Your comments will be reviewed directly by the BEA professionals who create and update the BEA WebLogic Adapter for SAP documentation.

In your e-mail message, please indicate that you are using the documentation for BEA WebLogic Adapter for SAP and the version of the documentation.

If you have any questions about this version of BEA WebLogic Adapter for SAP, or if you have problems using the BEA WebLogic Adapter for SAP, contact BEA Customer Support through BEA WebSUPPORT at www.bea.com. You can also contact Customer Support by using the contact information provided on the Customer Support Card which is included in the product package.

When contacting Customer Support, be prepared to provide the following information:

- Your name, e-mail address, phone number, and fax number
- Your company name and company address
- Your machine type and authorization codes
- The name and version of the product you are using
- A description of the problem and the content of pertinent error messages

Help Us to Serve You Better

To help our consultants answer your questions effectively, please be prepared to provide specifications and sample files and to answer questions about errors and problems.

The following tables list the specifications our consultants require.

Platform	
Operating System	
OS Version	
Product List	
Adapters	
Adapter Deployment	For example, JCA, or Integration Business Services Engine.
Container Version	

The following table lists components. Specify the version in the column provided.

Component	**Version**
Adapter	
EIS (DBMS/APP)	
HOTFIX / Service Pack	

The following table lists the types of Application Explorer. Specify the version (and platform, if different than listed previously) in the columns provided.

Application Explorer Type	**Version**	**Platform**
Swing		
Servlet		
ASP		

In the following table, specify the JVM version and vendor in the columns provided.

Version	**Vendor**

The following table lists additional questions to help us serve you better.

Request/Question	**Error/Problem Details or Information**
Provide usage scenarios or summarize the application that produces the problem.	
Did this happen previously?	
Can you reproduce this problem consistently?	
Any **change in the application environment**: software configuration, EIS/ database configuration, application, and so forth?	
Under what circumstance does the problem *not* occur?	
Describe the **steps** to reproduce the problem.	
Describe the **problem**.	
Specify the **error** message(s).	

The following table lists error/problem files that might be applicable.

XML schema
XML instances
Other input documents (transformation)
Error screen shots
Error output files
Trace and log files
Log transaction

Contents

CHAPTER 1

Introducing the BEA WebLogic Adapter for SAP

Topics:

The following section provides an overview of the BEA WebLogic Adapter for SAP.

Features of the BEA WebLogic Adapter for SAP

The BEA WebLogic Adapter for SAP is a remote function call adapter that provides a means to exchange real-time business data between SAP R/3 systems and other application, database, or external business partner systems. The adapter enables external applications for inbound and outbound processing with SAP.

The adapter uses XML messages to enable non-SAP applications to communicate and exchange transactions with SAP using one of the following two methods.

- **Event Adapter.** Applications use this capability if they require access to SAP data only when an SAP business event occurs.
- **Request/response.** Applications use this capability when they must initiate an SAP business event.

  If the request is for retrieving data from SAP, then the adapter sends the application a response message in the form of an XML document with the data embedded.

The BEA WebLogic Adapter for SAP provides:

- Support for bidirectional message interactions.
- The Servlet Application Explorer, a GUI tool that uses SAP object repository metadata to build XML schemas and Web services to handle adapter requests or event data.
- Support for Remote Function Calls (RFC), Business Application Programming Interfaces (BAPI), and Intermediate Documents (IDoc) interfaces to SAP.

SAP Certification

The BEA WebLogic Adapter for SAP provides state-of-the-art middleware solutions for SAP Basis and SAP Web application server-based systems. The adapter has earned the following three interface certifications which demonstrate that it promotes cost-effective and low-risk solutions.

- **CA-ALE certification.** Enhances electronic data interchange (EDI) subsystem interface with SAP Basis and SAP Web Application Server. Using direct program-to-program remote communication and transformation from non-SAP systems to SAP solution-based systems, the adapter expedites the conversion, import, and export of critical intermediate documents (IDocs).
- **CA-AMS certification.** Rapidly bridges SAP Basis and SAP Web Application Server data exchange with other applications through pure message delivery. As an ALE (Application Link Enabling) Message Handler, the adapter sends IDoc messages without a requirement for conversion from one or more SAP solution-based systems.

- **CA-XML certification.** Eases the communication between external middleware with SAP Basis and SAP Web Application Server over the Internet using XML, HTTP, or HTTPS. The adapter immediately transforms SAP solution specifications into XML for straight transfer into application subsystem repositories. The adapter directly receives and converts messages to be pulled or pushed into XML formats to or from SAP solution-based systems over the Internet.

Supported Versions and Platforms

The BEA WebLogic Adapter for SAP supports R/3 and R/3 Enterprise. The following SAP R/3 versions are supported by the BEA WebLogic Adapter for SAP:

- 4.6C
- 4.6D
- SAP R/3 Enterprise 47x100

  SAP R/3 Enterprise Version 47x100 is supported on the SAP Web Application Server Versions 6.20 and 6.40.
- SAP R/3 Enterprise 47x200

  SAP R/3 Enterprise Version 47x200 is supported on the SAP Web Application Server Versions 6.20 and 6.40.
- SAP Java Connector (SAP JCo) 2.1.6

  Note: For the current release status of the SAP Java Connector, refer to SAP Note #549268 in the SAP Service Marketplace.

In addition, the BEA WebLogic Adapter for SAP supports those SAP versions that are on the mainstream maintenance track as defined by the SAP Service Marketplace release strategy. To access this document, use the following URL:

`http://service.sap.com/releasestrategy/`

If the SAP R/3 version you are looking for is not listed in this table, please consult your BEA Customer Service Representative.

The following 32-bit platforms are supported by the BEA WebLogic Adapter for SAP:

- MS Windows 2000 with SP2, MS Windows 2003
- Sun Solaris 8 and Sun Solaris 9
- HP-UX 11
- IBM AIX 5
- Red Hat Enterprise Linux ES version 3.0
- TRU64

- OS/400
- Z/OS

For additional platform or operating system support information for the BEA WebLogic Adapter for SAP, contact Customer Support.

SAP Business Components

The BEA WebLogic Adapter for SAP is designed to provide standard access to SAP business components such as Remote Function Call (RFC) modules, BAPIs (Business Application Programming Interfaces), and IDocs (Intermediate Documents), that are used to support existing business processes.

These business components and methods are available to the adapter as requests of SAP and to the event adapter when SAP invokes its remote requests and work in the following ways:

Business Application Programming Interfaces (BAPIs) are interfaces within the business framework that are used to link SAP components to one another or to third-party components. BAPIs are called synchronously and return information.

Remote Function Call (RFC) Modules are SAP application interfaces that enable clients to invoke SAP technologies and receive responses.

Note: Depending on the release or service pack installed, certain RFCs may not exist in your particular SAP system. Therefore, the examples included in this documentation may not be relevant to your system. If this is the case, you should use the examples as a general reference for adapter functionality and choose an RFC that exists within your SAP application environment.

As described in SAP Release Note 109533, SAP Function Modules (RFCs) can be delivered with different release statuses. SAP supports only RFCs that are awarded with the Released for Customer status. There is no claim to the release independencies of the interfaces and the continued existence/functionality of the modules. For more information on the status of a specific function module, consult your SAP Service Marketplace.

Intermediate Documents (IDocs) are the “logical messages” that correspond to different business processes. They enable different application systems to be linked by a message-based interface. The IDoc type indicates the SAP format to use to transfer the data for a business transaction. An IDoc is a real business process in the form of an IDoc type that can transfer several message types. An IDoc type is described by the following components:

- **Control records.** A control record contains data that identifies the sender, the receiver, and the IDoc structure. An IDoc contains one control record.

- **Data records.** A data record consists of a fixed administration part and a data part (segment). The number and format of the segments can be different for each IDoc type.
- **Status records.** A status record describes the processing stages through which an IDoc passes.

  The following scenario is an example of IDoc functionality and its components:

  Purchase order number 4711 was sent to a vendor as IDoc number 0815. IDoc number 0815 is formatted in IDoc type ORDERS01 and has the status records "created" and "sent." The purchase order corresponds to the "logical" message ORDERS.

Integrating With SAP

You can use the BEA WebLogic Adapter for SAP to invoke an SAP business process, for example, add/update account, or you can use the adapter as part of an integration effort to connect SAP and non-SAP systems.

BAPIs and RFCs are called synchronously by the adapter and always return data (either technical error information or a well-formed response document). IDocs are processed asynchronously.

The adapter is bidirectional and can process an event in SAP by receiving RFCs and IDocs directly from SAP. The SAP system can be configured to send an IDoc or RFC to a logical system when a certain event occurs, in this case to the adapter. The output sent by SAP can be in any of the following forms:

- An RFC request, for example, RFC_SYSTEM_INFO.
- A BAPI request, for example, BAPI_COMPANYCODE_GETLIST.
- An IDoc.

For request processing, the BEA WebLogic Adapter for SAP can send requests to SAP using the BAPI, RFC, or IDoc interfaces.

The adapter quickly and easily integrates your SAP IDocs, RFCs, and BAPIs with mission critical SAP system applications and other enterprise applications. The benefits of the adapter include:

- Elimination of the requirement for custom coding.
- Consistent data representation.

  Provides a standard XML representation of event data and request/response documents for SAP.

  The developer is freed from the specific details of the SAP interface (BAPI, RFC, IDoc) and the specific configuration details of the target SAP system.

- Adherence to SAP ABAP serialization rules and SAP Interface Repository standards published by SAP AG.

Understanding Web Services and Java Connector Architecture Functionality

The following section describes how the BEA WebLogic Adapter for SAP can incorporate Web services and Java Connector Architecture technology.

Web Services

Web services allow SAP calls to be made across the Internet or an intranet, using specialized versions of the XML language that allow a developer to specify the parameters, connections methods, and remote calls and store them for reference in a repository. At runtime, a person, an interface, or another function, can read this repository and automatically invoke the service. Web services currently do not have industry standards for transactional behavior. Web services are useful when your function calls must be made across firewall boundaries. Using Web services, you can use functions provided by external providers, as long as you know the function interface.

Web Services Example

A Web service exposes the “cup” interface, which provides a teacup. The Acme Company exposes the “tea” Web service, which provides a brown liquid when the correct parameter “money” is provided. A cup of tea can be received by invoking the “tea” Web service and passing the “money” parameter. Additional components are not required to receieve tea via the “tea” Web service.

Java Connector Architecture

Java Connector Architecture (JCA) provides a reusable component model to build and deploy multi-tier applications that are platform and vendor-independent. JCA acts as a type of envelope or “container” that will allow the adapter to run inside the BEA WebLogic Server and connect to SAP and immediately return the results. JCA is useful when your SAP system resides within a local intranet or is accesed directly. JCA implements JAVA Connection and Transaction models. JCA requires a resource adapter to be physically deployed on the host application server to access the remote EIS system.

Using combinations of JCA and Web services is possible. For example, a JCA application can be invoked via a Web service or a Web service may be implemented inside a JCA container. The standards and protocols are still evolving.

JCA Example

An application server,"Table" implements the "tea" container, which contains "tea, gas heat, water, and cup". A JCA container can be deployed that contains tea in the "Table" application server. If any of the elements of "tea" container are missing, for example, "gas heat", then no tea is produced. The container must physically possess the "tea, gas heat, water, and cup" components to receive the tea.

Component Information for the BEA WebLogic Adapter for SAP

The BEA WebLogic Adapter for SAP works in conjunction with one of the following components:

- Integration Business Services Engine (iBSE)
- Enterprise Connector for J2EE™ Connector Architecture (JCA)

Component Information Roadmap

The following table lists the deployment component and the location of component information for the BEA WebLogic Adapter for SAP. A description of Application Explorer, the Integration Business Services Engine (iBSE), and the Enterprise Connector for J2EE Connector Architecture (JCA) follows the table

Deployed Component	**For more information, see**
Application Explorer	Chapters 3, 4, and 5 of this guide Appendix A of this guide when using Application Explorer inside WebLogic Workshop *BEA WebLogic ERP Adapter Installation and Configuration*
Integration Business Services Engine (iBSE)	*BEA WebLogic ERP Adapter Installation and Configuration*
Enterprise Connector for J2EE Connector Architecture (JCA)	*BEA WebLogic ERP Adapter Installation and Configuration*

Application Explorer

Application Explorer uses an explorer metaphor to browse the SAP system for metadata. The explorer enables you to create XML schemas and Web services for the associated object. In addition, you can create ports and channels to listen for events in SAP. External applications that access SAP through the BEA WebLogic Adapter for SAP use either XML schemas or Web services to pass data between the external application and the adapter.

The following versions of Application Explorer are available when deploying the adapter with BEA WebLogic Server:

- **Servlet.** Deployed as a Web application, this version is accessible through a Web browser. In addition, the servlet Application Explorer can be used with Integration Business Services Engine (iBSE) and Enterprise Connector for J2EE Connector Architecture (JCA). For more information, see the following chapters:
  - Chapter 3, *Creating XML Schemas for SAP*
  - Chapter 4, *Creating and Publishing Integration Business Services*
  - Chapter 5, *Configuring the Event Adapter for SAP*
- **Integrated Java Swing.** Tightly integrated within the BEA WebLogic toolset, the integrated Java Swing Application Explorer can be accessed directly from WebLogic WorkShop, where WSDL (Web Services Description Language) files generated from Integration Business Services and XML schemas can be shared as resources within a WebLogic WorkShop application. For more information, see Appendix A, *Using Application Explorer in BEA WebLogic Workshop to Create XML Schemas and Web Services*.

  Note: To use Application Explorer within WebLogic WorkShop, you must deploy the Integration Business Services Engine (iBSE). For more information, see the *BEA WebLogic ERP Installation and Configuration* documentation.

Integration Business Services Engine

The Integration Business Services Engine (iBSE) exposes—as Web services—enterprise assets that are accessible from adapters regardless of the programming language or the particular operating system.

iBSE simplifies the creation and execution of Web services when running:

- Custom and legacy applications.
- Database queries and stored procedures.
- Packaged applications.
- Terminal emulation and screen-based systems.
- Transactional systems.

Web services is a distributed programming architecture that solves Enterprise Application Integration (EAI) hurdles that other programming models cannot. It enables programs to communicate with one another using a text-based, platform and language independent message format called XML.

Coupled with a platform and language independent messaging protocol called SOAP (Simple Object Access Protocol), XML enables application development and integration by assembling previously built components from multiple Web services.

Enterprise Connector for J2EE Connector Architecture (JCA)

The Enterprise Connector for J2EE Connector Architecture (JCA) enables developers of JCA-compliant applications to deploy adapters as JCA resources. The connector is supported on J2EE-compliant application servers, such as your BEA WebLogic Server.

The Connector for JCA is distributed as a standard Resource Adapter Archive (RAR) for deployment to the application server. Thus, the connector can be used in systems that are non-compliant, although services such as pooled connections are not available.

CHAPTER 2

Configuring SAP Inbound Processing

Topics:

The following section describes how to configure your SAP system for inbound (client) processing.

Overview

SAP Remote Function calls require no system setup other than the Connection Target Parameters. If you do not intend to send IDocs to the SAP system, you may skip this chapter.

In inbound IDoc processing, the adapter reads an XML document and creates an SAP Standard format IDoc. The adapter can optionally transform incoming data into the standard format through XML transformation or SAP ALE transformation. After assembled as an IDoc, the file is sent to SAP for inbound processing. Multiple IDocs can also be assembled into a larger document for efficient processing. For more information, see the SAP documentation.

ALE IDocs used for transmission of Electronic Data Interchange (EDI) messages require information about the intended target and method of transmission stored on the BEA WebLogic Server.

An IDoc consists of a Header that contains sender, processing type, receiver and other information, and multiple data segments, which contain the information to be processed. To store the parameters for processing of the IDoc messages, SAP requires a "logical system" entry for each transmission system. The Logical System stores type information about the Partner and the kind of messages expected from the partner.

The Partner Profile defines the kind of message and the type of SAP function called to process that particular IDoc in an application. SAP also requires a filtering model, whether or not it is used for a particular message, called a Distribution Model that defines a message type and applies optional segment filters. After these are established, Inbound ALE/IDoc processing can begin.

The Control Section or reference structure file EDI_DC40 (defined in SAP R/3), must be completed and contains all the identifying information about the IDoc. The assembled header and data records are sent to SAP by the adapter. SAP does not require an incoming port to be specified. It takes the incoming RFC stream and assigns a port designation. In the Partner Profile, a function module must be identified to process the IDoc in the R/3 system.

Usually, IDocs are written directly to the database and slowly read by the application (for example, Purchasing for Purchase Orders). This can take time depending on the type of data and the application. The adapter can "post to the database and return" or "post and wait." This is defined in the Partner Profile on the host system. In either case, you may send a status IDoc message to obtain the status of your IDoc or use appropriate transaction codes in SAP to view the IDocs online.

You must perform the following steps to configure SAP for inbound IDoc processing:

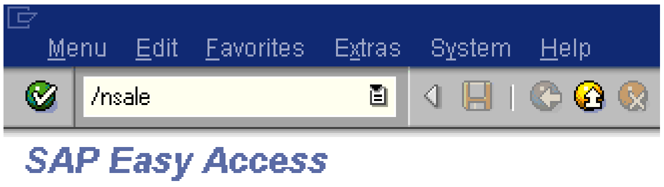
1. Configure a logical system.
2. Configure a distribution model.
3. Define an inbound partner profile.

Configuring a Logical System

In a distributed environment, each participating system must have a unique ID to avoid confusion. In SAP, the name of the logical system is used as the unique ID. This name is assigned explicitly to one client in an SAP system.

Procedure: How to Configure a Logical System

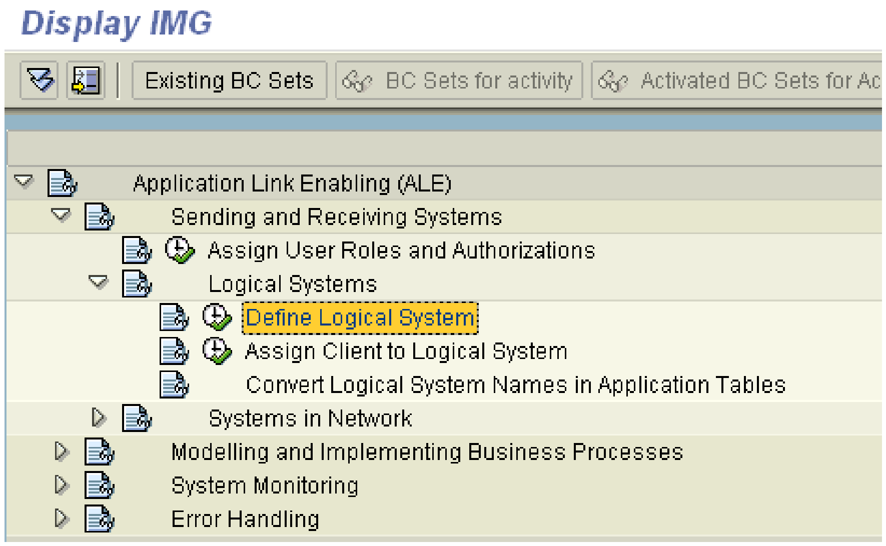
The following image shows the /nsale transaction in the field under the menu bar.



To configure a logical system:

1. Execute the *sale* transaction.

   The Display IMG window opens as shown in the following image.

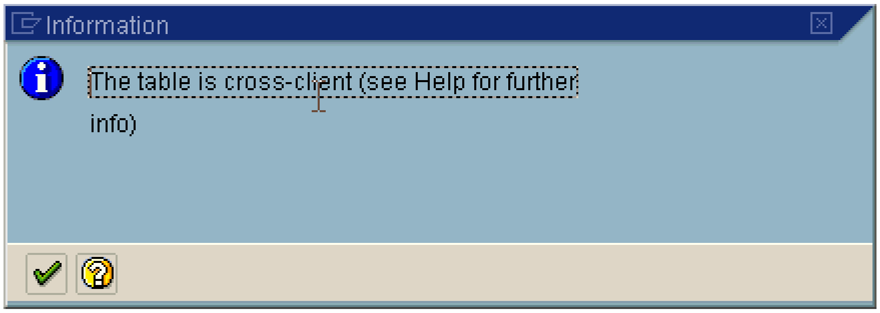

Display IMG
Existing BC Sets
BC Sets for activity
Activated BC Sets for Ac
Application Link Enabling (ALE)
Sending and Receiving Systems
Assign User Roles and Authorizations
Logical Systems
Define Logical System
Assign Client to Logical System
Convert Logical System Names in Application Tables
Systems in Network
Modelling and Implementing Business Processes
System Monitoring
Error Handling

   a. Expand *Sending and Receiving Systems* and then, *Logical Systems*.

   b. Select *Define Logical System*.

2. Click the *IMG - Activity* icon.

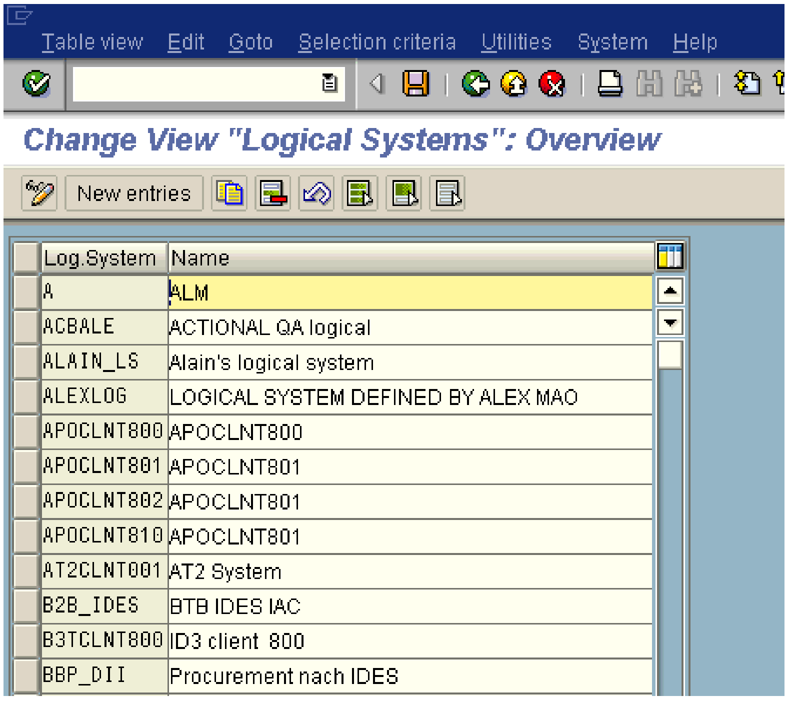
An information window appears that informs you that the table is cross-client as shown in the following image.



The table is cross-client (see Help for further info)

3. To continue, click the checkmark icon.

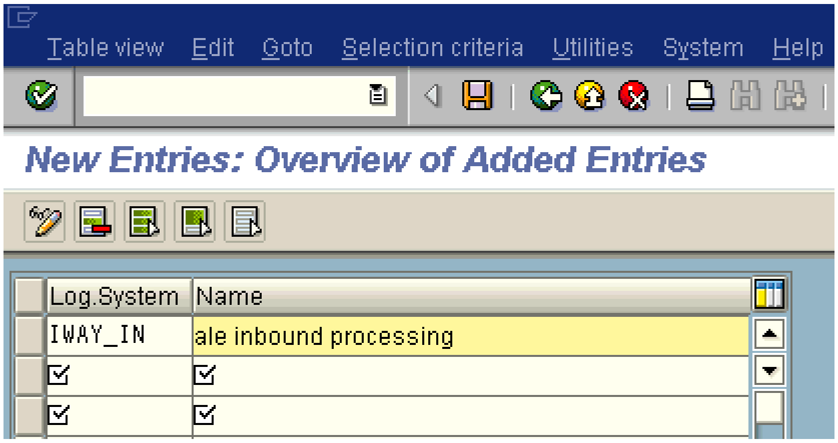
The Change View “Logical Systems” window opens with a list of logical systems and their names as shown in the following image.



Log.System	Name
A	ALM
ACBALE	ACTIONAL QA logical
ALAIN_LS	Alain's logical system
ALEXLOG	LOGICAL SYSTEM DEFINED BY ALEX MAO
APOCLNT800	APOCLNT800
APOCLNT801	APOCLNT801
APOCLNT802	APOCLNT801
APOCLNT810	APOCLNT801
AT2CLNT001	AT2 System
B2B_IDES	BTB IDES IAC
B3TCLNT800	ID3 client 800
BBP_DII	Procurement nach IDES

4. Click the *New entries* button.

The New Entries window opens where you can type information for the logical system and its corresponding name as shown in the following image.



New Entries: Overview of Added Entries

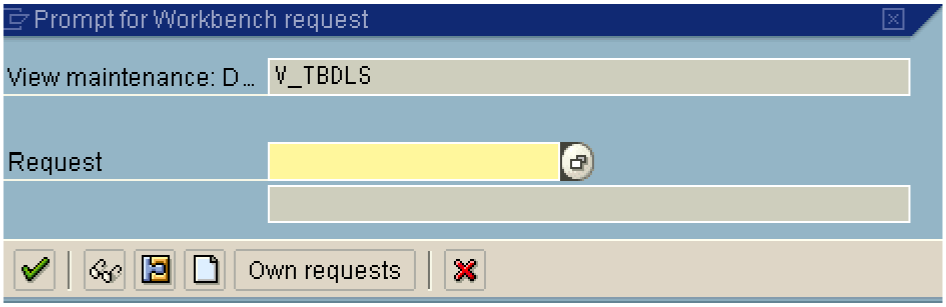
Log.System	Name
IWAY_IN	ale inbound processing
✓	✓
✓	✓

a. In the Log.System column, type the Logical System, for example, IWAY_IN.

b. In the Name column, type a corresponding description.

5. Click *Save*.

   The Prompt for Workbench request window opens as shown in the following image. It includes fields for View maintenance and Request as well as several buttons.



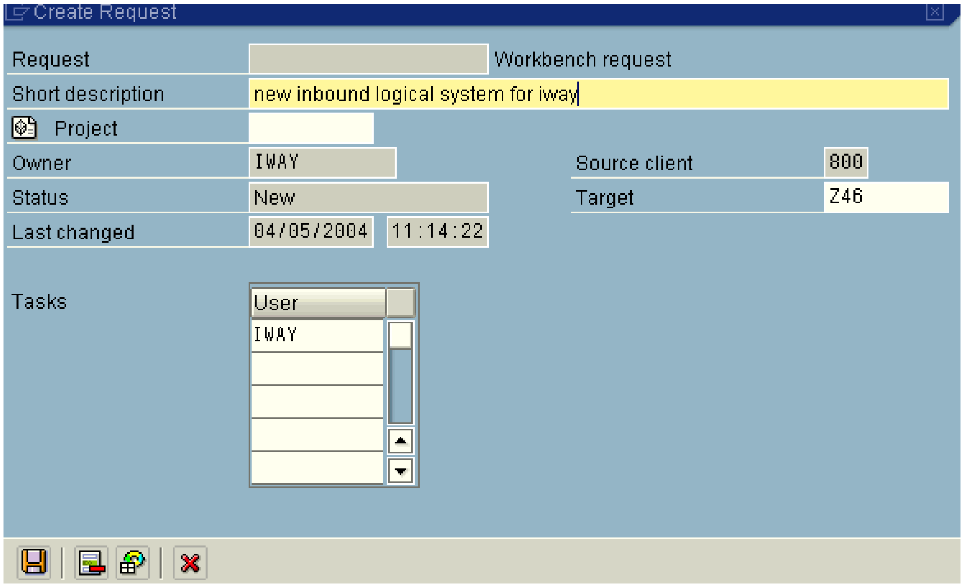
Prompt for Workbench request

View maintenance: D... V_TBDLS

Request

6. Click the *Create Request* icon.

The Create Request window opens as shown in the following image. It includes fields that are already populated (such as Owner, Status, Last Changed, Source client, and so forth), empty fields (such as Request and Project) in which to specify information about your request, and a Tasks list.



Request		Workbench request	
Short description	new inbound logical system for iway		
Project			
Owner	IWAY	Source client	800
Status	New	Target	Z46
Last changed	04/05/2004 11:14:22		

Tasks

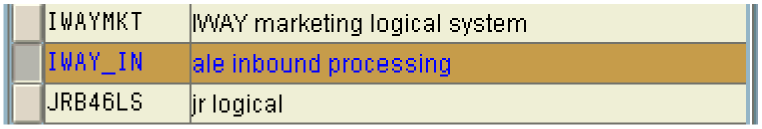
User
IWAY

a. In the Request field, type a name.

b. In the Short description field, type a brief description of your request.

7. Click *Save*.

The logical system you configured, for example, IWAY_IN, appears in the list as shown in the following image.



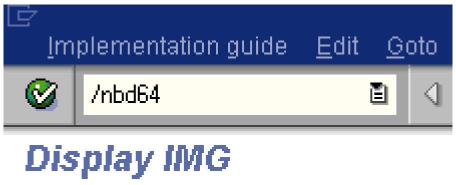
IWAYMKT	IWAY marketing logical system
IWAY_IN	ale inbound processing
JRB46LS	jr logical

Configuring a Distribution Model

A distribution model is used to describe the ALE message flow between logical systems. Business objects are distributed to connected recipients according to a unique distribution model that can contain rules of varying complexity depending on the type of business objects involved.

Procedure: How to Configure a Distribution Model

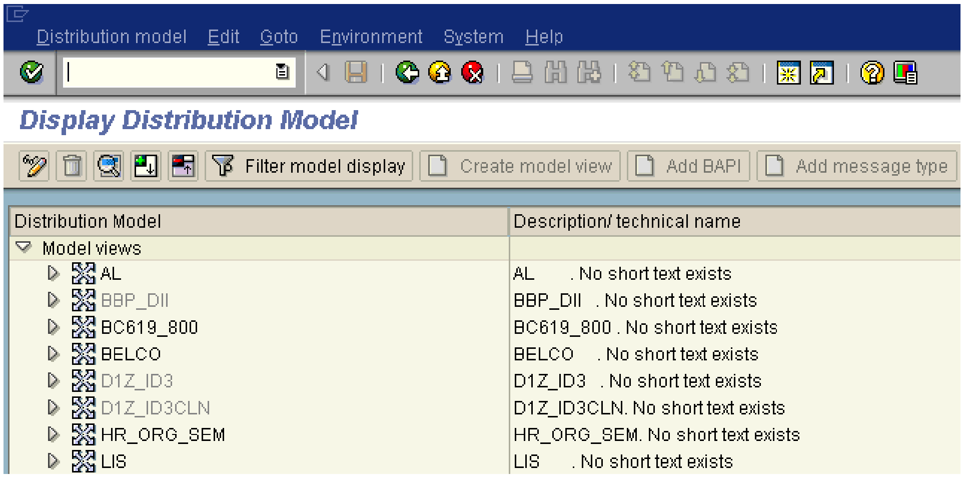
The following image shows the /nbd64 transaction in the field under the menu bar.



To configure a distribution model:

1. Execute the */bd64* transaction.

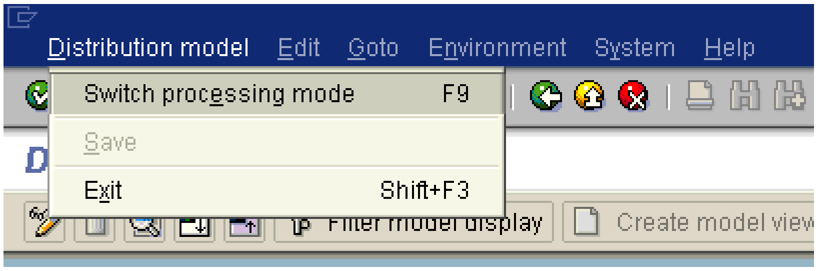
   The Display Distribution Model window opens and displays a list of available distribution models and their descriptions as shown in the following image.



Distribution Model	Description/ technical name
Model views	
AL	AL . No short text exists
BBP_DII	BBP_DII . No short text exists
BC619_800	BC619_800 . No short text exists
BELCO	BELCO . No short text exists
D1Z_ID3	D1Z_ID3 . No short text exists
D1Z_ID3CLN	D1Z_ID3CLN. No short text exists
HR_ORG_SEM	HR_ORG_SEM. No short text exists
LIS	LIS . No short text exists

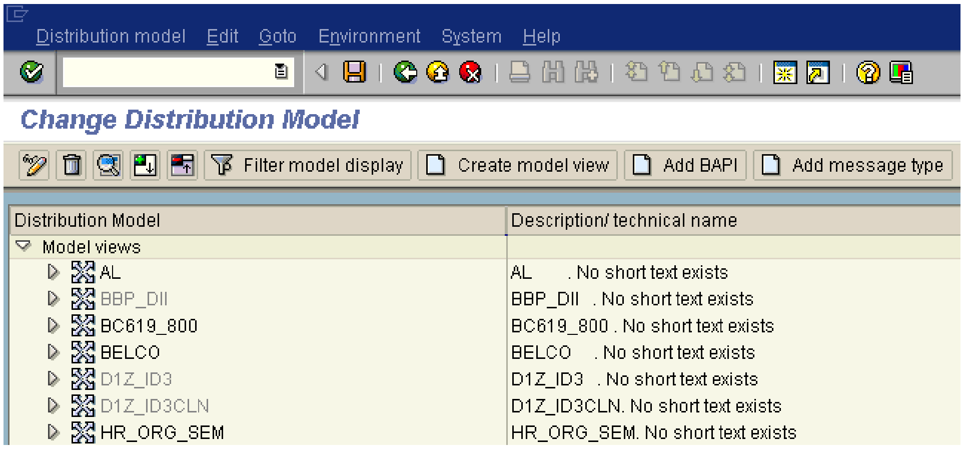
2. In the menu bar, click *Distribution model*.

   The Distribution model menu opens as shown in the following image.



3. Select *Switch processing mode.*

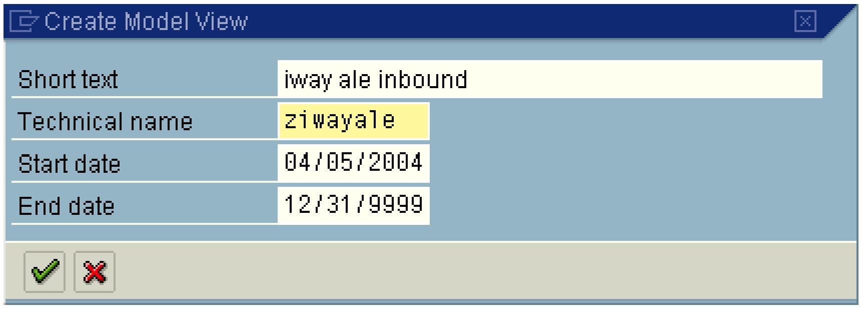
   The Display Distribution Model window switches to the Change Distribution Model. window as shown in the following image.


Distribution model Edit Goto Environment System Help
Change Distribution Model
Filter model display
Create model view
Add BAPI
Add message type

Distribution Model	Description/ technical name
Model views	
AL	AL . No short text exists
BBP_DII	BBP_DII . No short text exists
BC619_800	BC619_800 . No short text exists
BELCO	BELCO . No short text exists
D1Z_ID3	D1Z_ID3 . No short text exists
D1Z_ID3CLN	D1Z_ID3CLN. No short text exists
HR_ORG_SEM	HR_ORG_SEM. No short text exists

4. Click the *Create model view* button.

   The Create Model View window opens and includes fields for the name of your distribution model and for Start and End dates as shown in the following image.



Short text	iway ale inbound
Technical name	ziwayale
Start date	04/05/2004
End date	12/31/9999

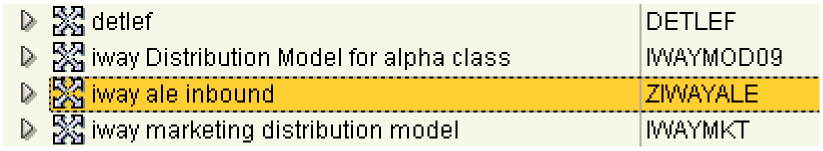
   a. In the Short text field, type a model view name, for example, iway ale inbound.

   b. In the Technical name filed, type a technical name, for example, ziwayale, which also serves as a description.

5. To enter the information, click the checkmark icon.

   You are returned to the main Change Distribution Model window.

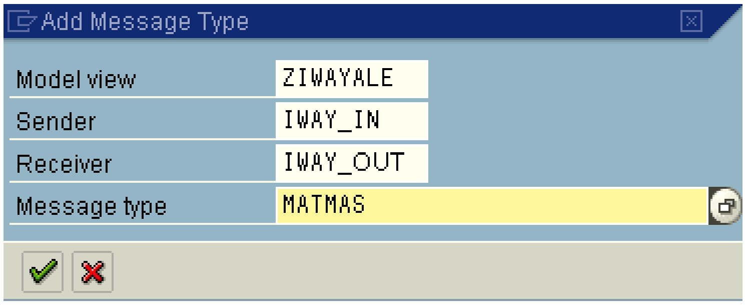
The distribution model you configured is now added to the list as shown in the following image.



detlef	DETLEF
iway Distribution Model for alpha class	IWAYMOD09
iway ale inbound	ZIWAYALE
iway marketing distribution model	IWAYMKT

6. Click the *Add message type* button.

   The Add Message Type window opens and includes fields where you can name and specify your message type.



Model view	ZIWAYALE
Sender	IWAY_IN
Receiver	IWAY_OUT
Message type	MATMAS

   a. In the Sender field, type the logical system you configured, for example, IWAY_IN.

   b. In the Receiver field, type the logical system you configured, for example, IWAY_OUT.

   c. In the Message type field, type the message type to use, for example, MATMAS.

      To browse from a list of available message types, you can click the icon to the right of the field.

7. To enter the information, click the checkmark.

   You are returned to the main Change Distribution Model window.

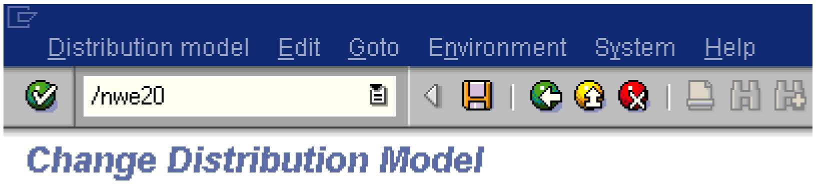
8. Click *Save*.

Defining a Partner Profile

Partner profiles are a requirement for data exchange. You define who can exchange messages with the SAP system using a specified port.

Procedure: How to Define a Partner Profile

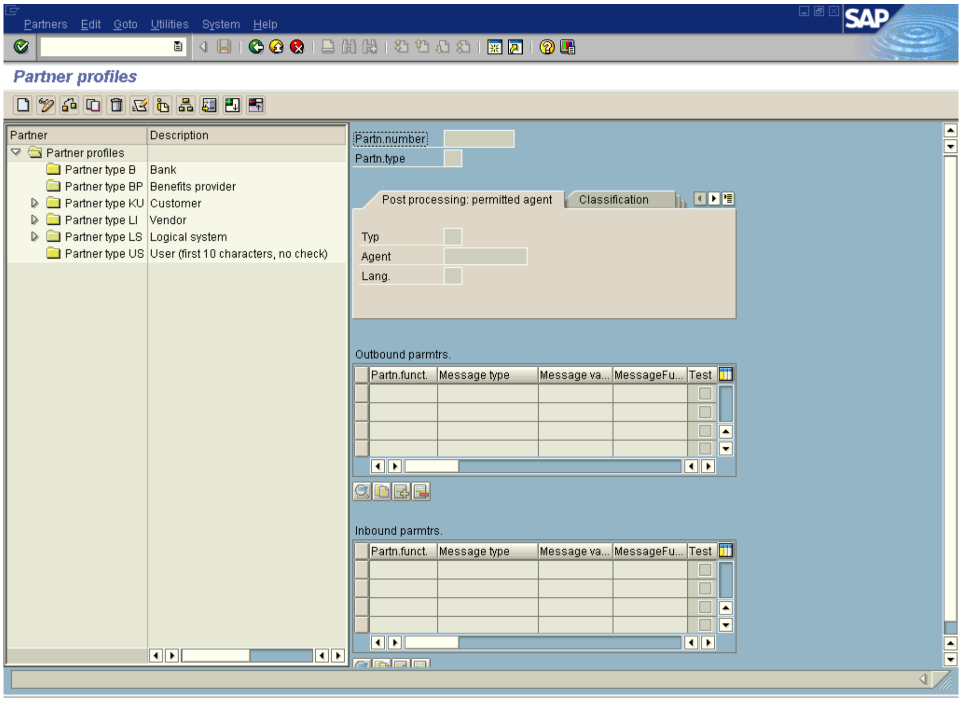
The following image shows the /nwe20 transaction in the field under the menu bar.


Distribution model Edit Goto Environment System Help
/nwe20
Change Distribution Model

To define a partner profile for a specific IDoc:

1. Execute the *we20* transaction.

   The Partner profiles window opens and displays two panes with information about the logical system as shown in the following image.

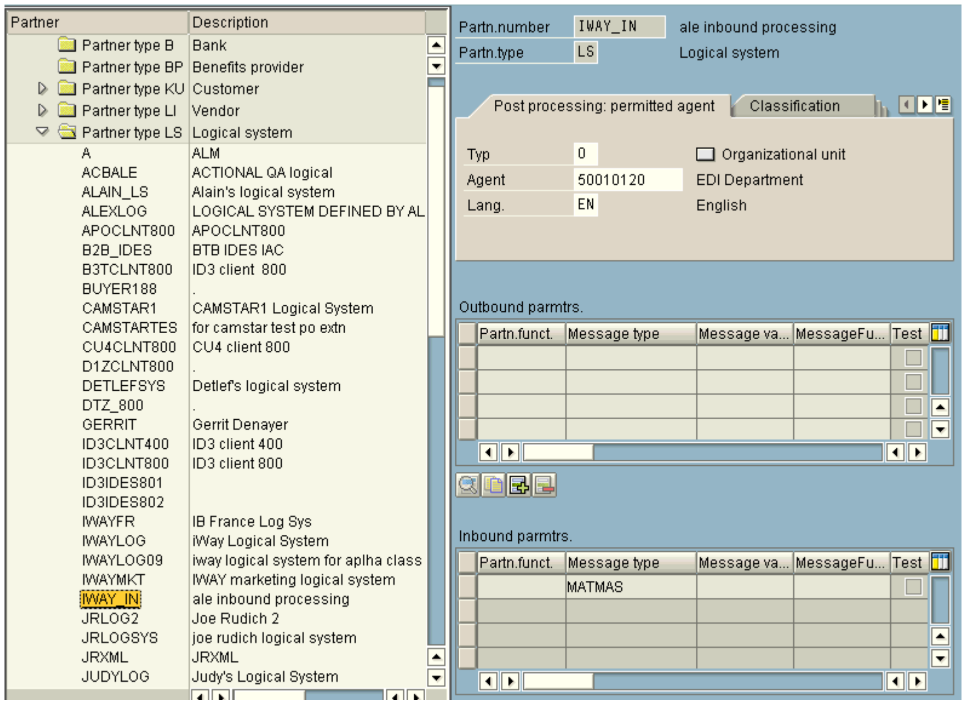


Partner	Description
Partner profiles	
Partner type B	Bank
Partner type BP	Benefits provider
Partner type KU	Customer
Partner type LI	Vendor
Partner type LS	Logical system
Partner type US	User (first 10 characters, no check)

2. In the left pane, expand *Partner type LS* and select the logical system you configured from the list, for example, IWAY_IN.

The right pane displays the details of the expanded folder including the logical system and type, language, and so forth, as shown in the following image.

Note: The Partn.number field refers to the name of the logical system.



Partner	Description
Partner type B	Bank
Partner type BP	Benefits provider
Partner type KU	Customer
Partner type LI	Vendor
Partner type LS	Logical system
A	ALM
ACBALE	ACTIONAL QA logical
ALAIN_LS	Alain's logical system
ALEXLOG	LOGICAL SYSTEM DEFINED BY AL
APOCLNT800	APOCLNT800
B2B_IDES	BTB IDES IAC
B3TCLNT800	ID3 client 800
BUYER188	.
CAMSTAR1	CAMSTAR1 Logical System
CAMSTARTES	for camstar test po extn
CU4CLNT800	CU4 client 800
D1ZCLNT800	.
DETLEFSYS	Detlef's logical system
DTZ_800	.
GERRIT	Gerrit Denayer
ID3CLNT400	ID3 client 400
ID3CLNT800	ID3 client 800
ID3IDES801	
ID3IDES802	
IWAYFR	IB France Log Sys
IWAYLOG	iWay Logical System
IWAYLOG09	iway logical system for aplha class
IWAYMKT	IWAY marketing logical system
IWAY_IN	ale inbound processing
JRLOG2	Joe Rudich 2
JRLOGSYS	joe rudich logical system
JRXML	JRXML
JUDYLOG	Judy's Logical System

Partn.number IWAY_IN ale inbound processing
Partn.type LS Logical system

Post processing: permitted agent | Classification

Typ	0	Organizational unit
Agent	50010120	EDI Department
Lang.	EN	English

Outbound parmtrs.

Partn.funct.	Message type	Message va...	MessageFu...	Test

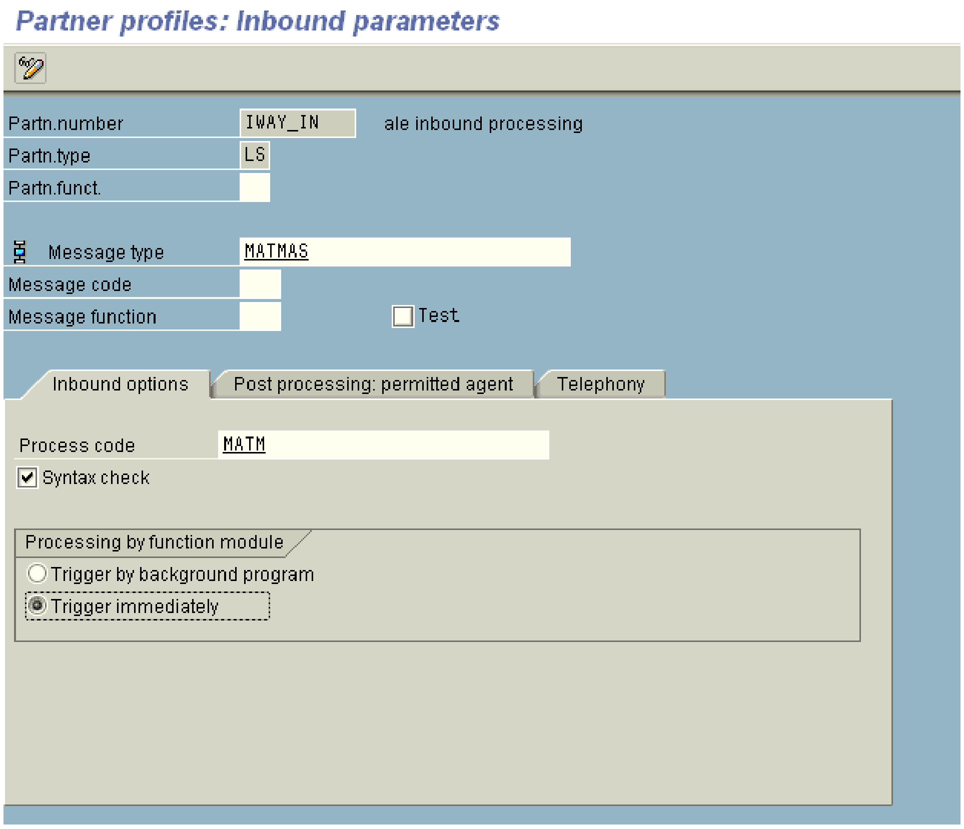
Inbound parmtrs.

Partn.funct.	Message type	Message va...	MessageFu...	Test
	MATMAS			

3. Click *Save*.

4. From the Inbound parameters table in the lower right, click the *Create inbound parameter* icon.

The Partner profiles: Inbound parameters window opens as shown in the following image.


Partner profiles: Inbound parameters

Partn.number: IWAY_IN — ale inbound processing
Partn.type: LS
Partn.funct.:

Message type: MATMAS
Message code:
Message function:
[ ] Test

Process code: MATM
[✓] Syntax check

Processing by function module
( ) Trigger by background program
(●) Trigger immediately

a. In the Message type field, type the message type to use, for example, MATMAS.

   The Inbound options tab is selected by default.

b. In the Process code field, enter the process code you want to use, for example, MATM.

c. In the Processing by function module area, select one of the following options:

   Trigger by background program. In this case, the BEA WebLogic Adapter for SAP writes IDocs to the SAP database, which are processed immediately.

   Trigger immediately. In this case, the BEA WebLogic Adapter for SAP waits for the SAP system to process IDocs. This can take from one to fifteen minutes.

5. Click *Save*.

CHAPTER 3

Creating XML Schemas for SAP

Topics:

- Overview
- Starting Servlet Application Explorer
- Establishing a Target for SAP
- Viewing Application System Objects
- Creating an XML Schema

The following section describes how to create XML schemas for SAP business objects using Application Explorer.

The functionality of Application Explorer is standard for any deployment type. This section uses the Java™ servlet implementation of Application Explorer to provide examples.

For information on running Application Explorer in WebLogic Workshop, see Appendix A, *Using Application Explorer in BEA WebLogic Workshop to Create XML Schemas and Web Services*.

Overview

The BEA WebLogic Adapter for SAP enables the processing of SAP BAPIs, RFCs, and IDocs.

External applications that access SAP through the adapter use either XML schemas or Web services to pass data between the external application and the adapter. You can use Servlet Application Explorer to create the required XML schemas and Web services.

AE is a Web application running within a servlet container that is accessible through a Web browser. For more information on installing and configuring the Servlet Application Explorer, see the *BEA WebLogic ERP Adapter Installation and Configuration* documentation.

SAP must be installed, configured, and available for client access. Application Explorer need not reside on the same system as the application system being accessed, but network access is required.

Starting Servlet Application Explorer

Before you can use the Servlet Application Explorer to browse metadata and create XML schemas, you must start the BEA WebLogic Server.

Procedure: How to Start BEA WebLogic Server on Windows

To start the BEA WebLogic Server on Windows:

1. Click the Windows Start menu.
2. Select *Programs*, *BEA WebLogic Platform 8.1*, *User Projects*, *your domain for iWay*, and then, click *Start Server*.

Procedure: How to Start BEA WebLogic Server on UNIX

To start the BEA WebLogic Server on UNIX or from a command line:

1. Display a prompt.
2. At the prompt, type the following:

   ```
   BEA_HOME\user_projects\domains\DOMAIN_NAME\startWebLogic.cmd
   ```

   where:

   `BEA_HOME`

   Is the directory where BEA WebLogic is installed.

   `DOMAIN_NAME`

   Is the domain you are using for iWay.

Procedure: How to Start Servlet Application Explorer

To start Application Explorer:

1. Enter the following URL in your browser window:

   `http://`*`hostname`*`:`*`port`*`/iwae/index.html`

   where:

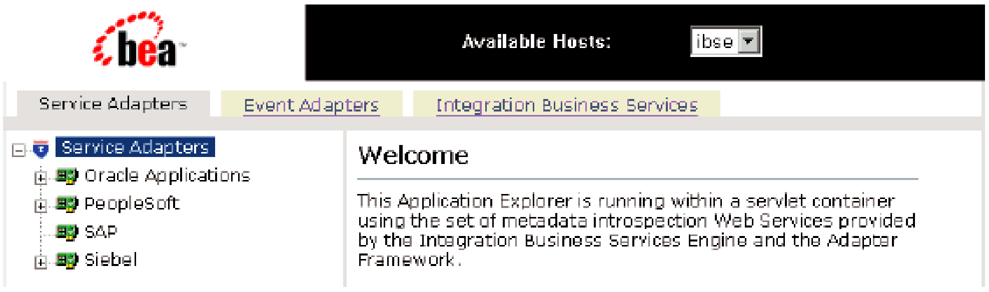
   `hostname`

   Is the name of the machine where your application server is running.

   `port`

   Is the port for the domain you are using for BEA. The port for the default domain is 7001.

   After you start Application Explorer, the following Welcome window opens, showing the Service Adapters, Event Adapters, and Integration Business Services tabs. The Service Adapters node is highlighted in the left pane.

   This Application Explorer is running within a servlet container using the set of metadata introspection Web Services provided by the Integration Business Services Engine and the Adapter Framework.

   The Available Hosts drop-down menu in the upper right lists the Connector for JCA or Servlet iBSE instance you can access.

2. For more information on adding instances, see *BEA WebLogic ERP Adapter Installation and Configuration*.

You are now ready to create new targets for SAP.

Establishing a Target for SAP

To browse SAP business objects, you must create a target for the system you intend to use. The target serves as your connection point and is automatically saved after you create it. You must establish a connection to this system every time you start Application Explorer or after you disconnect from the system.

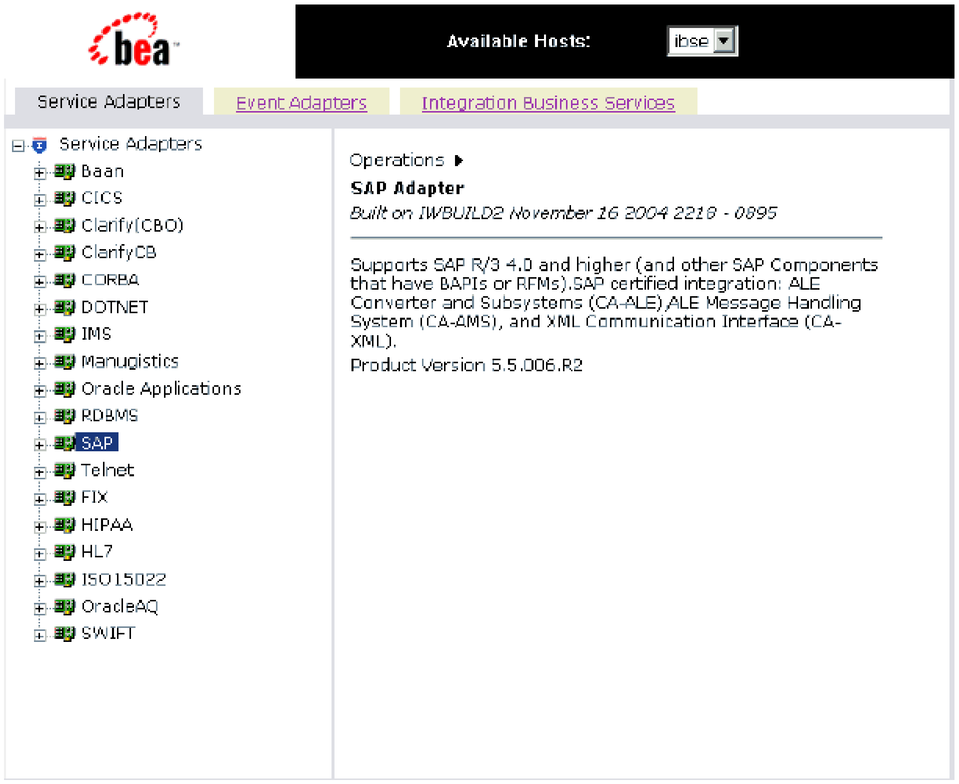
A list of supported application systems appears in the left pane of Application Explorer. The list is based on the adapters that you installed and have licenses to use.

Creating a New Target

To connect to an SAP system, you must define a new target. The target holds your logon parameters for the SAP system.

Procedure: How to Create a New Target

The following image shows a window with a navigation pane on the left that lists supported adapters. The right pane displays information about a selected adapter.



SAP Adapter

Built on IWBUILD2 November 16 2004 2218 - 0895

Supports SAP R/3 4.0 and higher (and other SAP Components that have BAPIs or RFMs).SAP certified integration: ALE Converter and Subsystems (CA-ALE),ALE Message Handling System (CA-AMS), and XML Communication Interface (CA-XML).

Product Version 5.5.006.R2

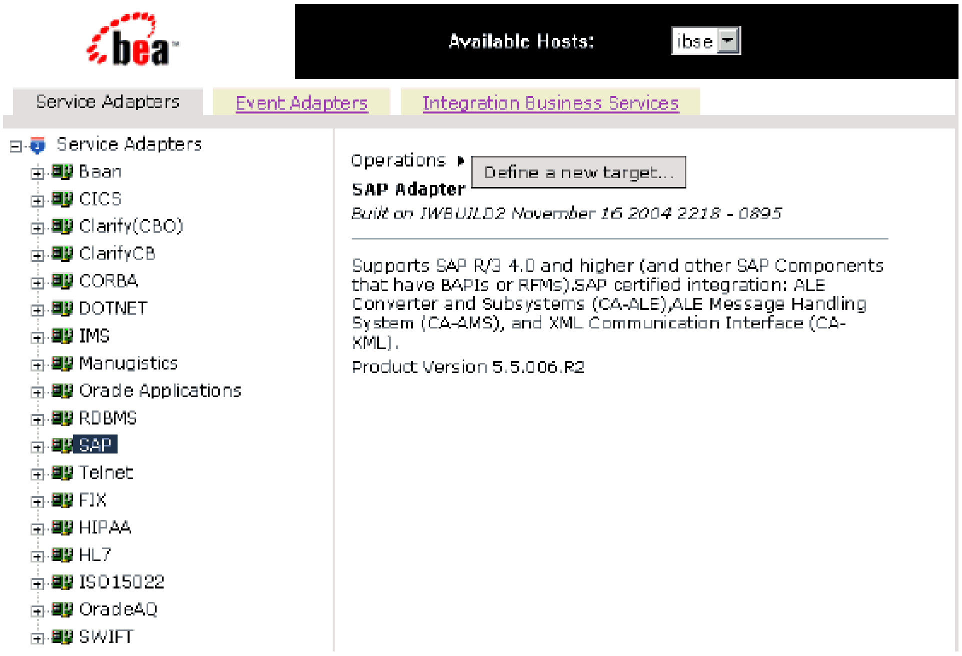
To create a new target:

1. In the left pane, click the *SAP* node.

   Descriptive information (for example, title and product version) about the BEA WebLogic Adapter for SAP appears in the right pane.

2. In the right pane, move the pointer over *Operations*.

The Define a new target menu option appears as well as title and product version information for the adapter in the right pane as shown in the following image.



SAP Adapter

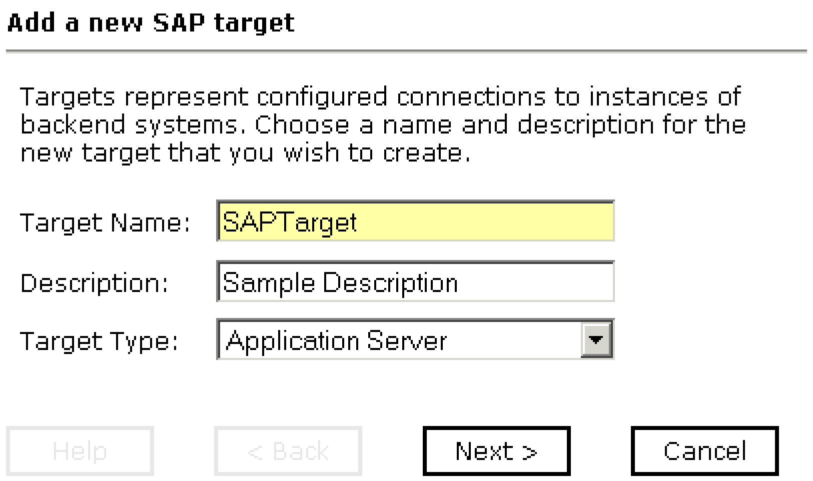
Built on IWBUILD2 November 16 2004 2218 - 0895

Supports SAP R/3 4.0 and higher (and other SAP Components that have BAPIs or RFMs).SAP certified integration: ALE Converter and Subsystems (CA-ALE),ALE Message Handling System (CA-AMS), and XML Communication Interface (CA-XML).

Product Version 5.5.006.R2

3. Select *Define a new target*.

   The Add a new SAP target pane opens on the right as shown in the following image.



Add a new SAP target

Targets represent configured connections to instances of backend systems. Choose a name and description for the new target that you wish to create.

Target Name: SAPTarget

Description: Sample Description

Target Type: Application Server

   a. In the Target Name field, type a name for the target, for example, SAPTarget.

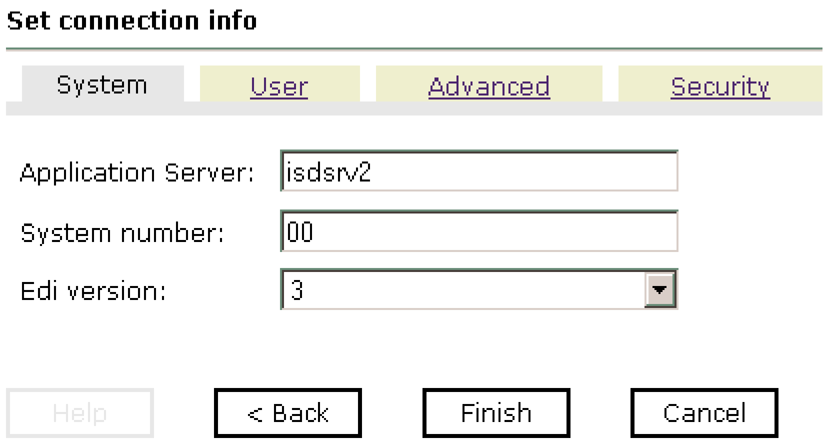
b. In the Target Description field, type a brief description (optional).

c. From the Target Type drop-down list, select the type of target to connect to.

   The default value is Application Server.

4. Click *Next*.

   The Set connection info pane opens on the right. The following tabs are available: System, User, Advanced, and Security. The System tab is active as shown in the following image.



Application Server: isdsrv2

System number: 00

Edi version: 3

Note: The SAP connection parameters are consistent with those found in your SAP system. For more information on parameter values that are specific to your SAP configuration, consult your SAP system administrator.

The following tabs are available:

- System (Required)
- User (Required)
- Advanced
- Security

The System (required) tab enables you to provide the application server name, system number, and EDI version for the SAP system to which you are connecting.

a. In the Application Server field, type the host name or IP address for the computer that is hosting the SAP application.

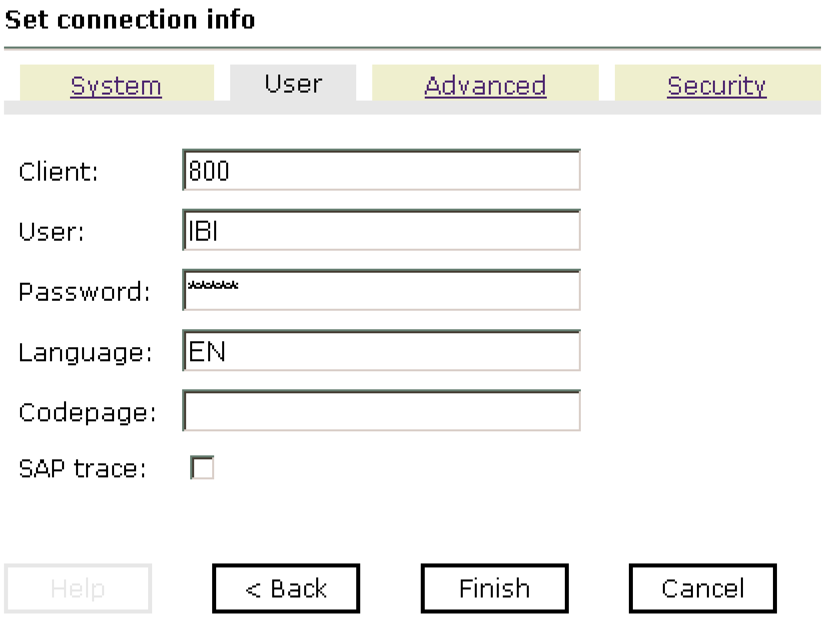
b. In the System Number field, type the system number defined to SAP for client communications.

- **c.** From the EDI Version drop-down list, select the Electronic Data Interchange (EDI) document version you are using with the BEA WebLogic Adapter for SAP.

  Version 3 is the default value.

5. Click the *User* tab.

The User (required) tab enables you to provide authentication information for the SAP system to which you are connecting.



Client: 800
User: IBI
Password: ******
Language: EN
Codepage:
SAP trace: ☐

- **a.** In the Client field, type the client number defined for the SAP application for client communications.
- **b.** In the User field, type a valid user ID for the SAP application.
- **c.** In the Password field, type a valid password for the SAP application.
- **d.** In the Language field type a language key.

  EN (English) is the default.
- **e.** In the Codepage field, type a character code page value.
- **f.** To enable traces, select the *SAP trace* check box.

After you provide information for the System and User tabs, you complete the basic SAP target configuration. However, you can specify additional parameters in the Advanced and Security tabs.

The Advanced tab contains the following fields:

- **Connection pool size.** This field is used to specify the number of client connections in a pool you want to make available to SAP for Web service calls.

  A default connection pool size of 1 is available by default. If you want to use a connection pool size that is greater than 1, enter the value in this field.

  Connections to an R3 server take up valuable resources on both the client and the remote server. The section of SAP documentation "Memory Management (BC-CST-MM)" explains in detail the resources required on the SAP system. A user logon also consumes processing time. You can create a pool of connections that will be continually recycled to minimize the resource and time constraints. The size of the connection pool will be determined by the size of your SAP server, available memory, maximum logons and the size of your client Java Virtual Machine. In estimating the size of the pool, you may calculate pool size by the amount of server resources to be consumed, the number and size of the documents to be received, and the size of your Java Virtual Machine.
- **Connection pool name.** Enter a name for your connection pool only if you specified a connection pool size that is greater than 1.
- **BAPI Exception Handling.** If your application is Java centric, you can select Throws Exception so that code components may catch the exception and react accordingly. If your application is document based, you can select Creates Document to have an XML document created that contains the Java exception. It is up to your application to read the XML document and obtain the error.
- **Commit with Wait.** All SAP Business Objects that change data must commit work to the database. Some BAPI's developed in version 3.1 of the R/3 system, use an internal commit behavior and their commit behavior cannot be changed by the adapter. As soon as they are called, they commit the work they have done. BAPI's developed since release 3.1 use the external commit method. The adapter issues a commit command, and the commit is put in the database que. If there is some application error in the first part of the commit, the error message "Posting could not be carried out" will be returned, and the adapter will roll back the transaction. If in writing to the database a database error occurs, a short dump will be issued in the database records of SAP, but no message will be returned to the adapter about the failure. If a high degree of accuracy is needed in your application, select "With Wait", and the adapter will wait until all records have been physically written to the database before returning from the function call. The "Commit With Wait" has a definite performance impact on your adapter performance, use it with deliberation. The commit behavior of BAPI's is found in SAP documentation under "Bapi Programming Guide and Reference (CA-BFA)".

  This option is disabled by default.

The Security tab contains the following fields:

- **Logon ticket (SSO2).** If you are using a Secure Network Communications (SNC) adapter with SAP, enter the name of the SSO2 logon ticket you are using.
- **Logon ticket (X509).** If you are using an SNC adapter with SAP, enter the name of the X.509 logon ticket you are using.

  Note: Depending on the SAP system release, logins using Single-Sign-On (SSO) or X.509 certificates are being supported.

  For SSO specify the user to be $MYSAPSSO2$ and pass the base64 encoded ticket as the passwd parameter.

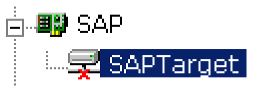
  For X509 specify the user to be $X509CERT$ and pass the base64 encoded certificate as the passwd parameter

  For more information regarding SSO or X.509 configuration, see your SAP system documentation.
- **SNC mode.** By default, SNC is disabled. To enable SNC, select 1 from the drop-down list.
- **SNC partner.** Enter the name of the RFC server or message server (load balancing) that provides the SNC services.
- **SNC level.** Select the version of the SNC library from the drop-down list.
- **SNC name.** Enter the name of the SNC library you are using.
- **SNC library path.** Enter the path to the SNC library.

SNC provides protection for the communication links between the distributed components of an R/3 System. Using SNC, SAP R/3 can support products which adhere to the GSS-API Version 2 standard. SNC supports application level (end-to-end security), Smartcard authentication, and single sign-on.

6. After you provide all the required information for your target, click *Finish*.

   The SAP target (SAPTarget) appears below the sap node in the left pane as shown in the following image. You are now ready to connect to your SAP target.



Connecting to a Target

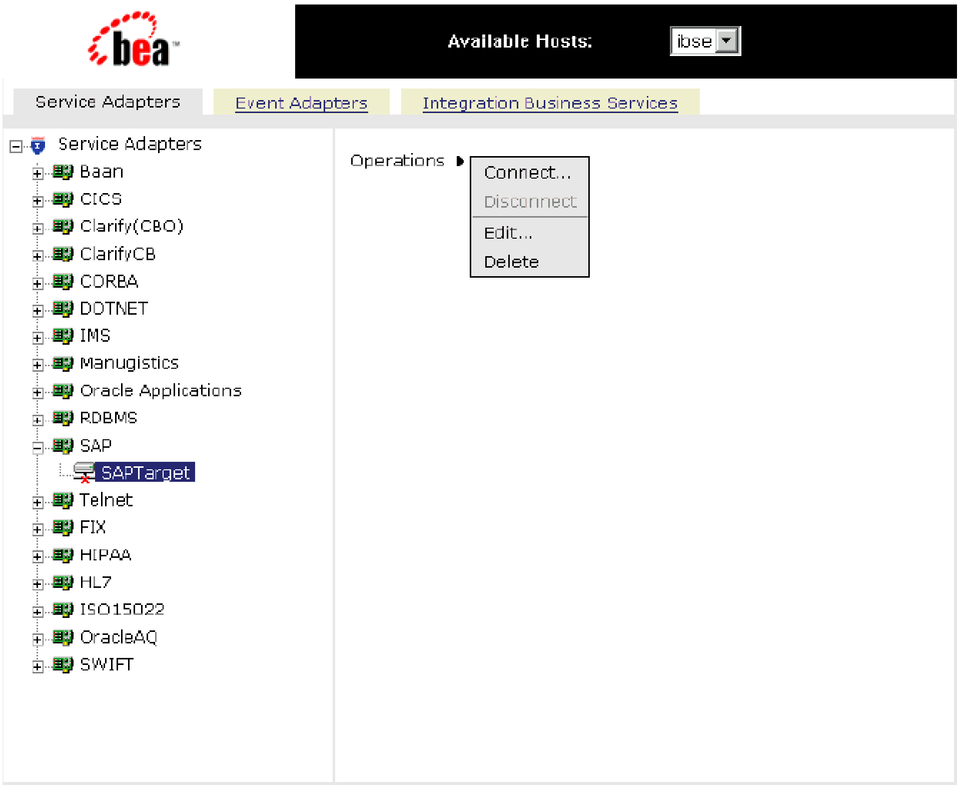
To connect to SAP, you use a target you defined, for example, the one in the previous procedure, SAPTarget.

Procedure: How to Connect to a Target

To connect to a target:

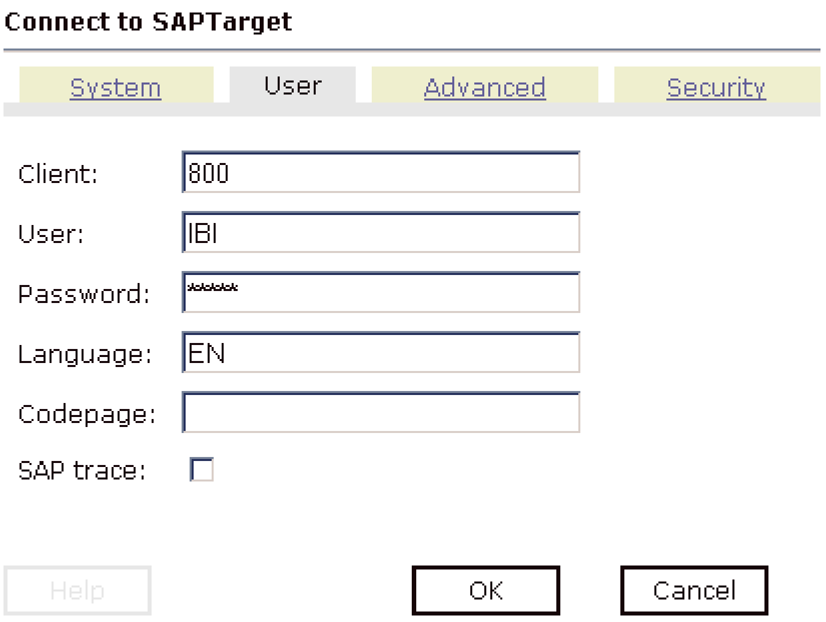
1. In the left pane, expand the sap node and select the target you defined, for example, SAPTarget.
2. In the right pane, move the pointer over *Operations.*

   The following image shows the target, with a red 'x', selected in the left pane. In the right pane, the Operations menu appears in its expanded form.



3. Select *Connect*.

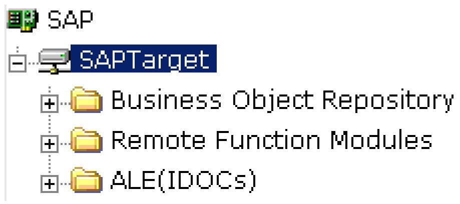
   The following graphic shows the Connect to SAPTarget pane that opens on the right, with fields to enter a client, a user, a password, a language, a code page, and an SAP trace.


Connect to SAPTarget

Client:	800
User:	IBI
Password:	******
Language:	EN
Codepage:	
SAP trace:	☐

4. In the Password field, type a valid password and click *OK*.

   The SAPTarget node in the left pane changes (the red 'x' disappears) to reflect that a connection was made as shown in the following image.



5. Expand the *SAPTarget* node.

   The following SAP business objects appear:

   - Business Object Repository
   - Remote Function Modules
   - ALE (IDocs)

Disconnecting From a Target

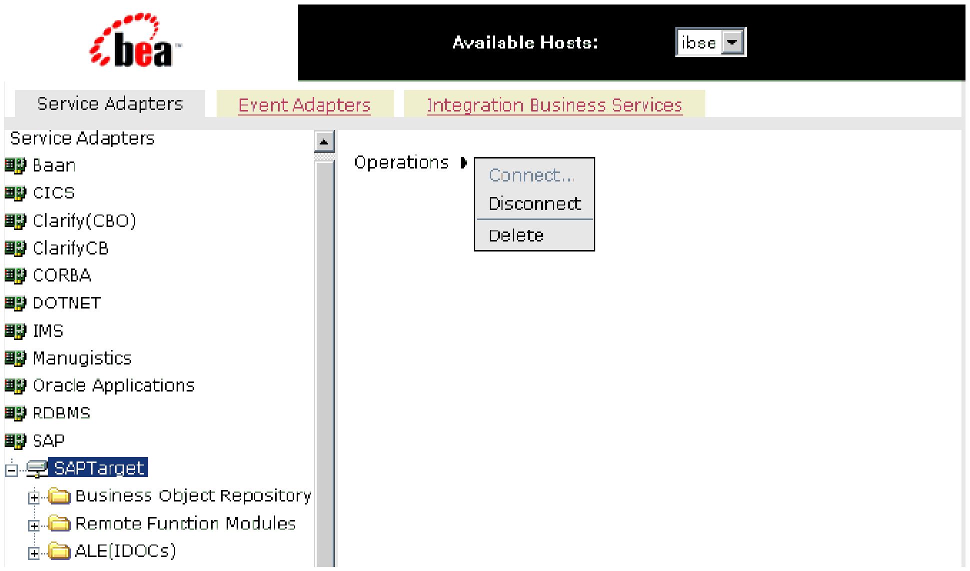
Although you can maintain multiple open connections to different application systems, it is a good practice to close connections when you are not using them.

Procedure: How to Disconnect From a Target

To disconnect from a target:

1. From the left pane, click the target, for example, SAPTarget, to which you are connected.
2. In the right pane, move the pointer over *Operations.*

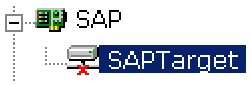
   The following image shows the target selected in the left pane. In the right pane, the Operations menu appears expanded to display options.



3. Select *Disconnect*.

   Disconnecting from the application system drops the connection, but the node remains.

   In the left pane, the SAPTarget node changes to reflect that a connection was closed (a red 'x' appears) as shown in the following image.



Modifying a Target

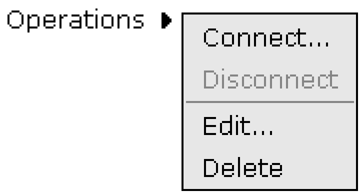
After you create a target for SAP using Servlet Application Explorer, you can edit the information that you provided previously.

Procedure: How to Edit a Target

To edit a target:

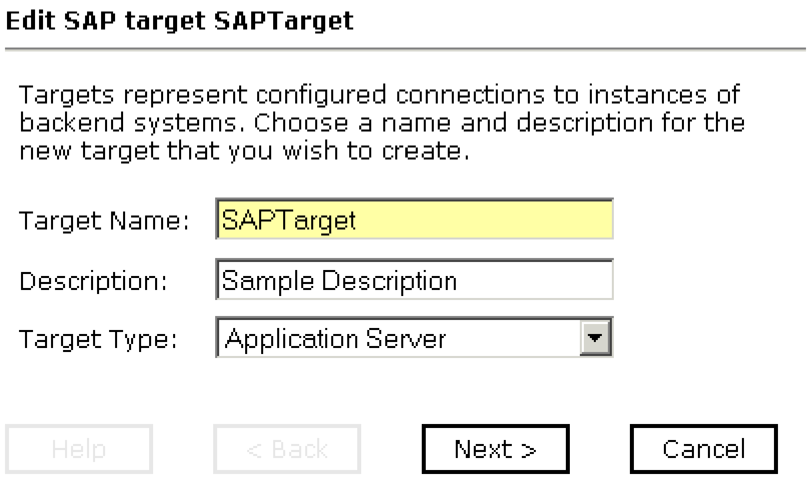
1. In the left pane, click the target, for example, SAPTarget.

   The Operations menu appears in the right pane, as shown in the following image.



2. Move the pointer over *Operations* and select *Edit*.

   The Edit pane opens on the right with the target name, a description and a target type selected from the drop-down list as shown in the following image.


Edit SAP target SAPTarget

Targets represent configured connections to instances of backend systems. Choose a name and description for the new target that you wish to create.

Target Name: SAPTarget

Description: Sample Description

Target Type: Application Server

3. Modify the connection information.
4. To continue modifying additional information, click *Next*.
5. When you are finished making all of your edits, click *Finish*.

Deleting a Target

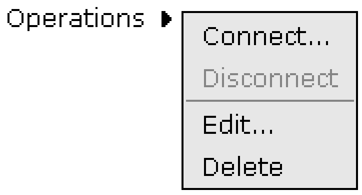
In addition to closing a target, you can delete a target that is no longer required. You can delete it whether or not it is closed. If open, the target automatically closes before it is deleted.

Procedure: How to Delete a Target

To delete a target:

1. In the left pane, click the target, for example, SAPTarget.

   The Operations menu appears in the right pane, as shown in the following image.



2. In the right pane, move the pointer over *Operations*.
3. Select *Delete*.

   A confirmation dialog box opens, asking if you want to delete the target.
4. To delete the target you selected, click *OK*.

   The SAPTarget node disappears from the left pane.

Viewing Application System Objects

After you are connected to SAP, Application Explorer enables you to explore and browse business object metadata. For example, Application Explorer enables you to view SAP BAPI, RFC, and iDoc metadata stored in the SAP Business Object repository.

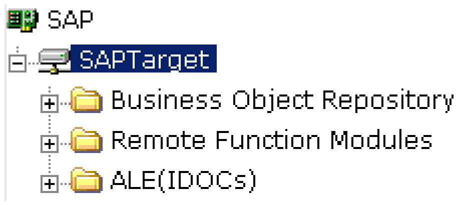
Note: Depending on the release or service pack installed, certain RFCs, for example, RFC_CUSTOMER_GET, may not exist in your particular SAP system. Therefore, the examples included in this documentation may not be relevant to your system. If this is the case, you should use the examples as a general reference for adapter functionality and choose an RFC that exists within your SAP application environment.

Procedure: How to View Application System Objects

To view application system objects:

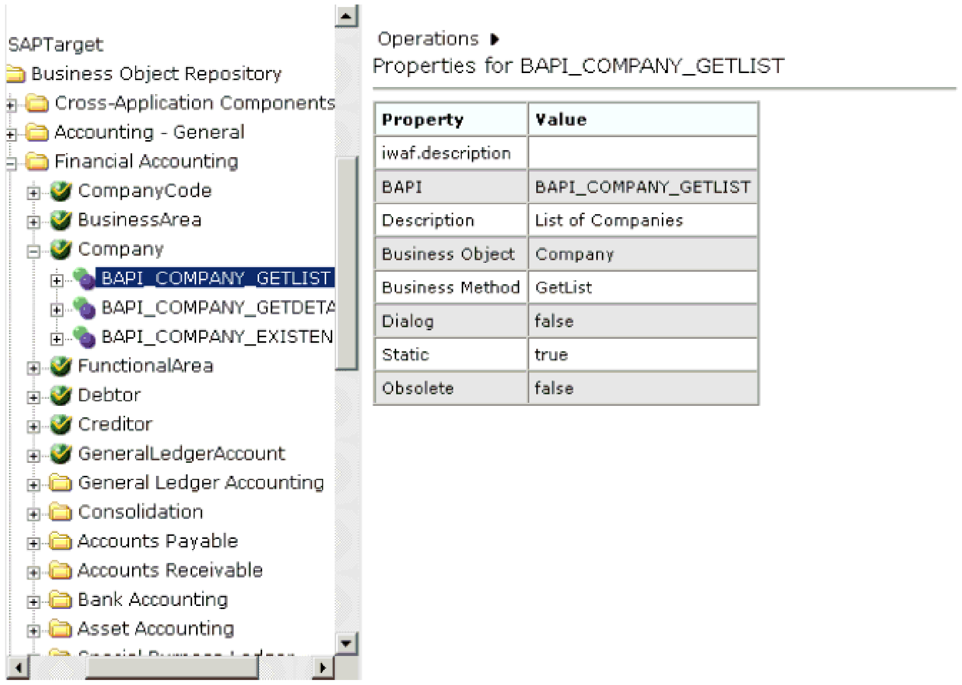
1. Click the icon to the left of the target name, for example, SAPTarget.

This expands the target to expose the available application system objects as shown in the following image.



2. To expand the desired SAP repository node, click the icon to the left of the repository name, for example, Business Object Repository.
3. In the list under Business Object Repository, click the icon next to *Financial Accounting*.

   A list of business objects related to Financial Accounting appears in the left pane. In the right pane, the collapsed Operations menu and a table listing properties and values for the BAPI method named BAPI_COMPANY_GETLIST appears as shown in the following image.

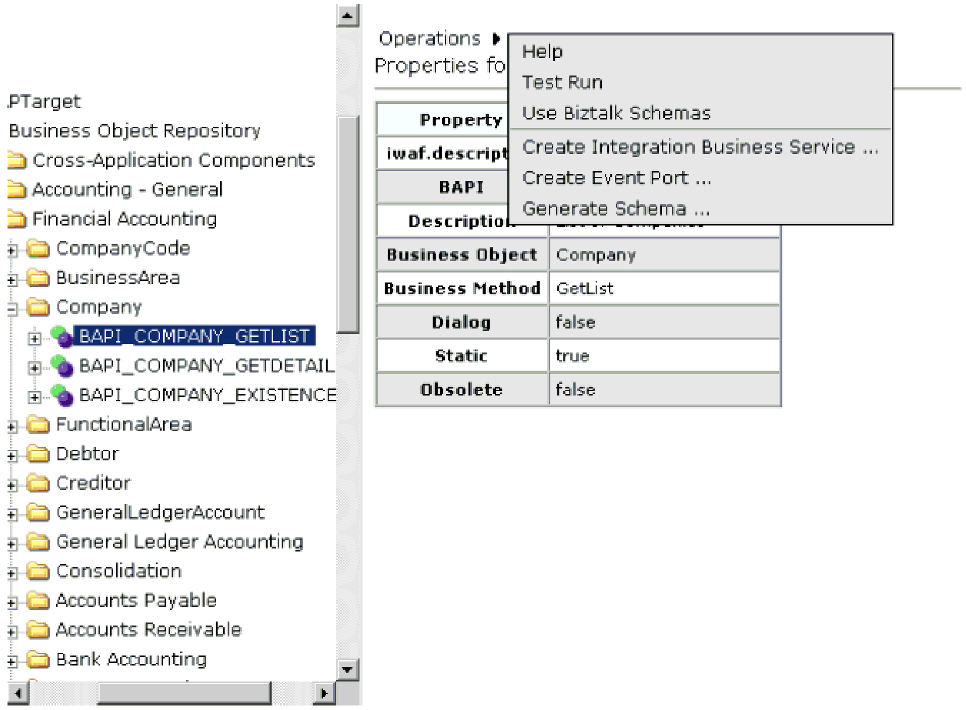


Property	Value
iwaf.description	
BAPI	BAPI_COMPANY_GETLIST
Description	List of Companies
Business Object	Company
Business Method	GetList
Dialog	false
Static	true
Obsolete	false

   a. Scroll down and click the icon next to the *Company* business object.
   b. Click the icon next to the BAPI method called *BAPI_COMPANY_GETLIST.*

4. In the right pane, move the pointer over *Operations* to view the context menu.

The following image shows the Operations menu expanded over the table listing properties and values for the BAPI method named BAPI_COMPANY_GETLIST in the right pane.



Property	
iwaf.descript	
BAPI	
Descriptio	
Business Object	Company
Business Method	GetList
Dialog	false
Static	true
Obsolete	false

The following options are available from the context menu:

- **Help** provides information about BAPI, RFC, and IDoc usage.
- **Test Run** simulates running the selected RFC or BAPI with sample data you provide.
- **Use Biztalk Schemas** sets the default for schema output that is used by Microsoft Biztalk.
- **Create Integration Business Services** creates Web services for the SAP business object you selected.
- **Create Event Port** creates a port to be used for SAP event handling.
- **Generate Schema** generates XML request and response schemas for the SAP business object you selected.

Creating an XML Schema

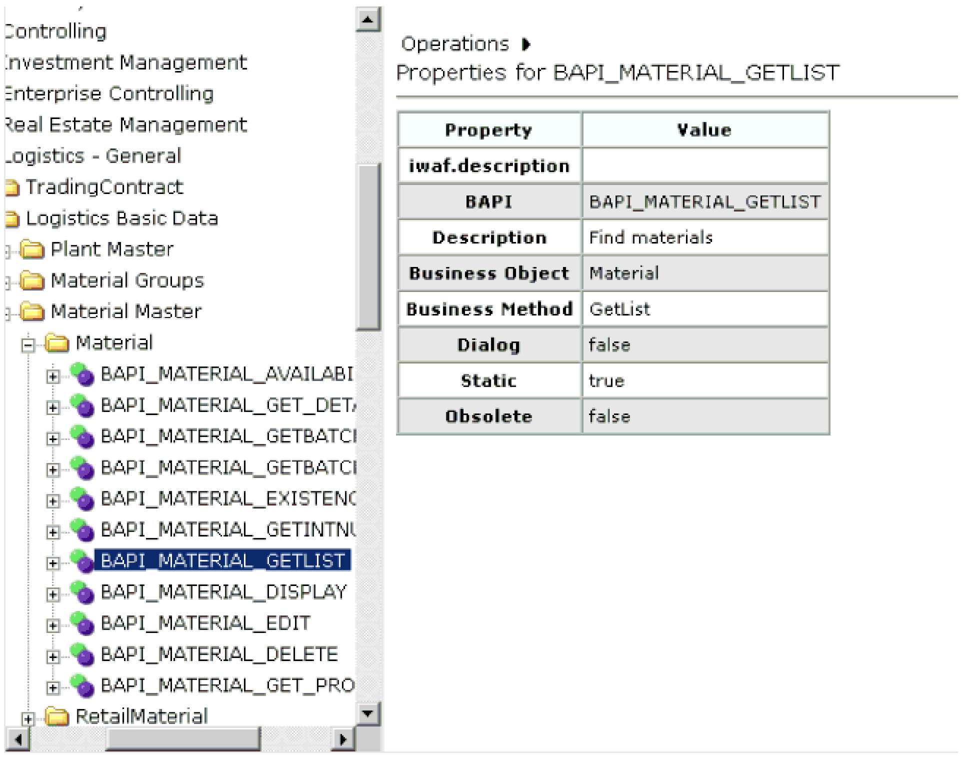
After you browse the SAP business object repository, you can generate XML request and response schemas for the object you wish to use with your adapter.

Procedure: How to Create XML Schemas

To create XML request and response schemas for the SAP BAPI method called BAPI_MATERIAL_GETLIST:

1. In the Business Object Repository, select the *GetList* method.

   In the right pane, the collapsed Operations menu and a table listing properties and values for the BAPI method named BAPI_MATERIAL_GETLIST appears as shown in the following image.



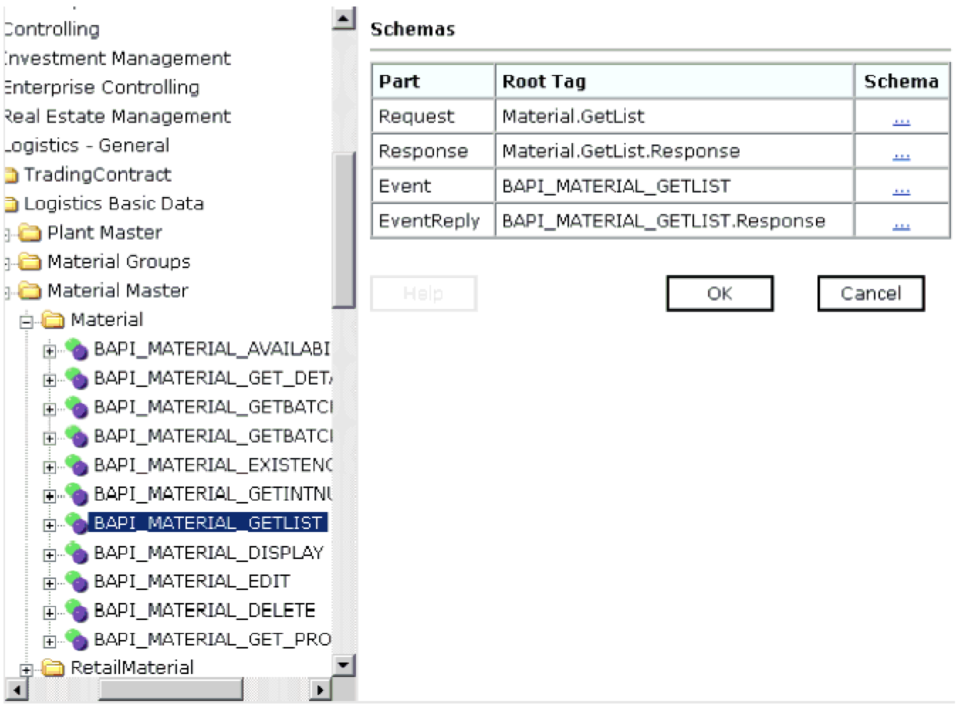
Properties for BAPI_MATERIAL_GETLIST

Property	Value
iwaf.description	
BAPI	BAPI_MATERIAL_GETLIST
Description	Find materials
Business Object	Material
Business Method	GetList
Dialog	false
Static	true
Obsolete	false

2. In the right pane, move the pointer over *Operations* and select *Generate Schema* from the menu.

   Request, response, and event schemas are created for your business object.

The following image shows the Schemas pane that opens on the right with a table that lists and defines the root tag for each schema and provides hyperlinks to click to view each schema.

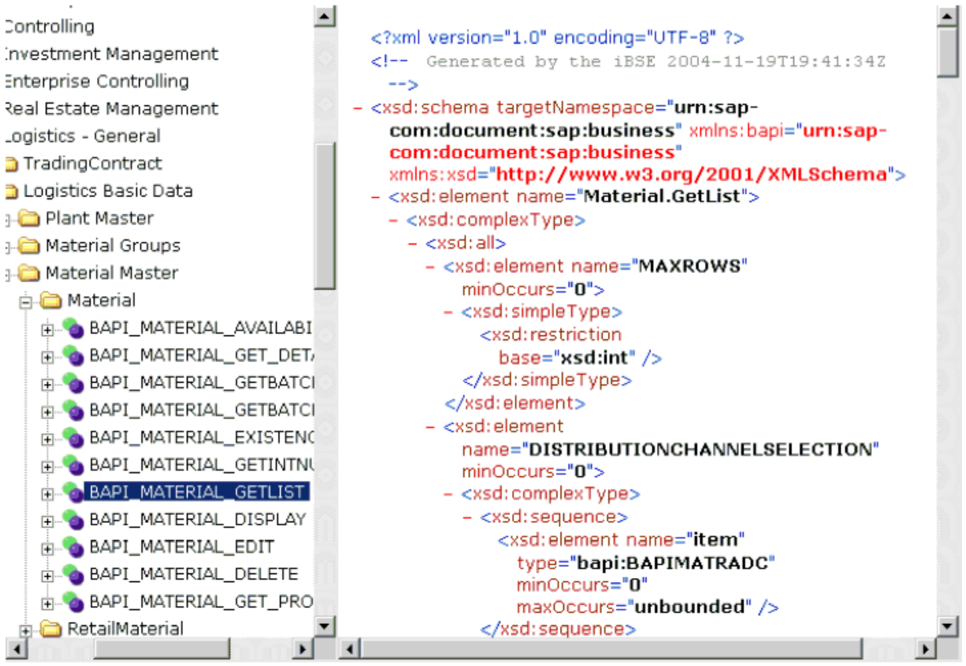


Schemas

Part	Root Tag	Schema
Request	Material.GetList	...
Response	Material.GetList.Response	...
Event	BAPI_MATERIAL_GETLIST	...
EventReply	BAPI_MATERIAL_GETLIST.Response	...

3. Click the hyperlink associated with the type of schema you want to view.

For example, if you click the hyperlink for the Request schema, the schema appears in the right pane as shown in the following image.



```
<?xml version="1.0" encoding="UTF-8" ?>
<!-- Generated by the iBSE 2004-11-19T19:41:34Z
  -->
- <xsd:schema targetNamespace="urn:sap-
    com:document:sap:business" xmlns:bapi="urn:sap-
    com:document:sap:business"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  - <xsd:element name="Material.GetList">
    - <xsd:complexType>
      - <xsd:all>
        - <xsd:element name="MAXROWS"
            minOccurs="0">
          - <xsd:simpleType>
              <xsd:restriction
                base="xsd:int" />
            </xsd:simpleType>
          </xsd:element>
        - <xsd:element
            name="DISTRIBUTIONCHANNELSELECTION"
            minOccurs="0">
          - <xsd:complexType>
            - <xsd:sequence>
                <xsd:element name="item"
                  type="bapi:BAPIMATRADC"
                  minOccurs="0"
                  maxOccurs="unbounded" />
              </xsd:sequence>
```

4. To return to the previous window, click the *Back* button on your Web browser.

   After you browse the list of business objects on your SAP system, you can create Integration Business Services. For more information, see Chapter 4, *Creating and Publishing Integration Business Services*.

   After the schemas are created, you also can create events. For more information, see Chapter 5, *Configuring the Event Adapter for SAP*.

CHAPTER 4

Creating and Publishing Integration Business Services

Topics:

This section describes how to create and publish Integration Business Services using Application Explorer.

The functionality of Application Explorer is standard for any deployment type. This section uses the Java™ servlet implementation of Application Explorer to provide examples.

For information on running Application Explorer in WebLogic Workshop, see Appendix A, *Using Application Explorer in BEA WebLogic Workshop to Create XML Schemas and Web Services*.

Understanding Integration Business Services

Servlet Application Explorer provides Web developers with a simple, consistent mechanism for extending the capabilities of the BEA WebLogic Adapter for SAP. The Integration Business Services Engine (iBSE) exposes functionality as Web services. It serves as a gateway to heterogeneous back-end applications and databases.

A Web service is a self-contained, modularized function that you can publish and access across a network using open standards. It is the implementation of an interface by a component and is an executable entity. For the caller or sender, a Web service can be considered as a "black box" that may require input and delivers a result. Web services integrate within an enterprise as well as across enterprises on any communication technology stack, whether asynchronous or synchronous, in any format.

After you browse the SAP business object repository and create an XML schema for the object, you can generate an Integration Business Service for the object you wish to use with your adapter.

Creating Integration Business Services

The following topics describe how to create Integration Business Services, including how to generate WSDL (Web Services Description Language) from a Web service.

Creating Business Services With Application Explorer

The following procedure describes how to create Integration Business Services using Servlet Application Explorer. The procedure uses the SAP BAPI method called BAPI_MATERIAL_GETLIST as an example and returns a list of materials.

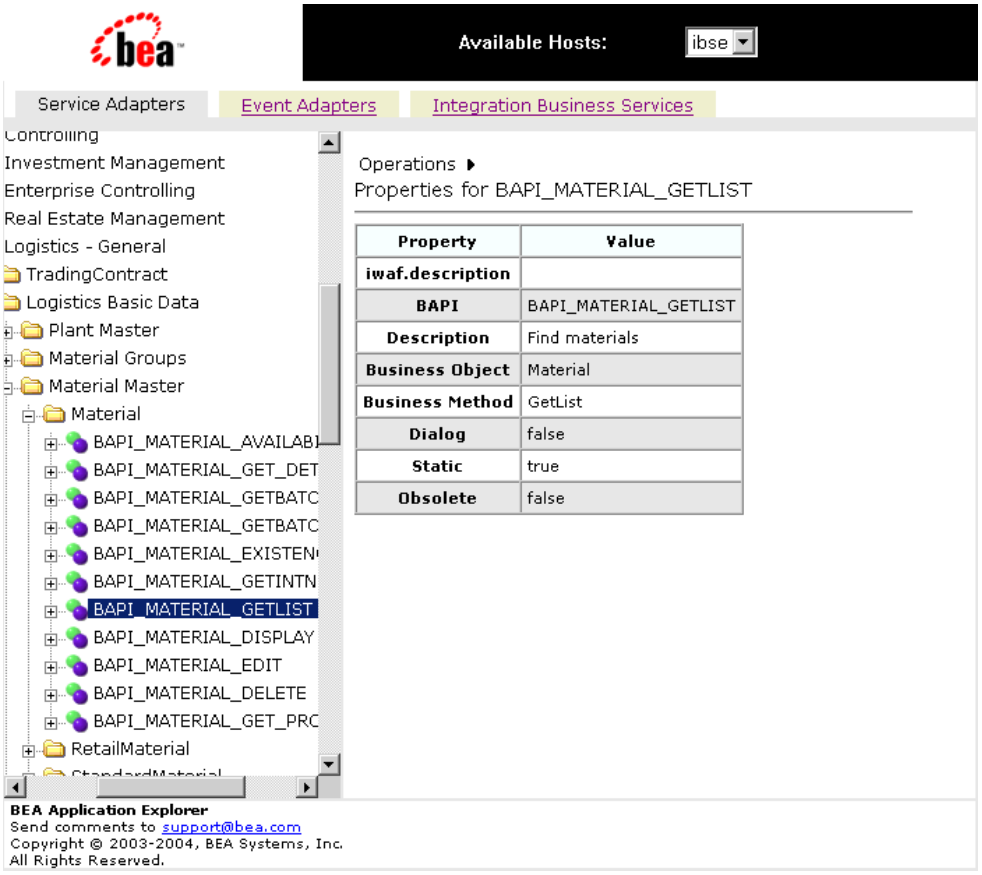
Note: If you want your Web service to use connection pooling, you must specify connection pooling information when connecting or reconnecting to your SAP target. For more information on connection pooling, see Chapter 3, *Creating XML Schemas for SAP*.

Procedure: How to Create Integration Business Services

To create Integration Business Services:

1. From the Business Object Repository, select the *BAPI_MATERIAL_GETLIST* method.

The following image shows the Operations menu in the right pane and a table listing properties and values for the BAPI method called BAPI_MATERIAL_GETLIST.



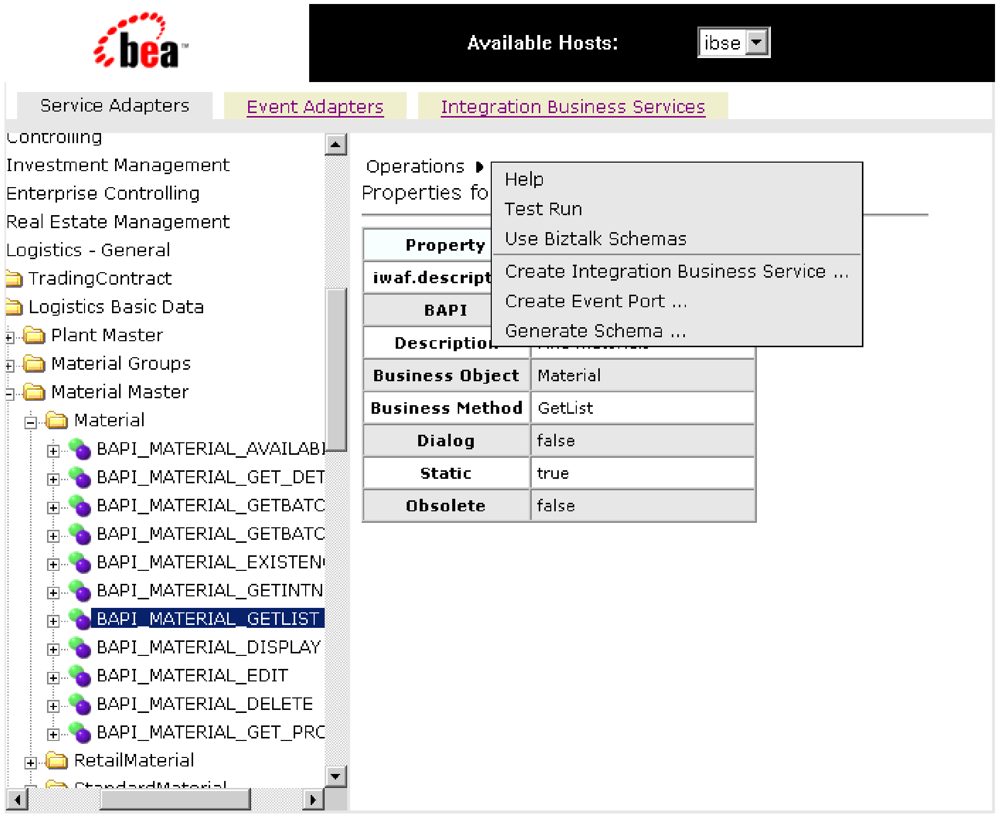
Properties for BAPI_MATERIAL_GETLIST

Property	Value
iwaf.description	
BAPI	BAPI_MATERIAL_GETLIST
Description	Find materials
Business Object	Material
Business Method	GetList
Dialog	false
Static	true
Obsolete	false

BEA Application Explorer
Send comments to support@bea.com
Copyright © 2003-2004, BEA Systems, Inc.
All Rights Reserved.

2. In the right pane, move the pointer over *Operations*.

The Operations menu expands as shown in the following image to display options.

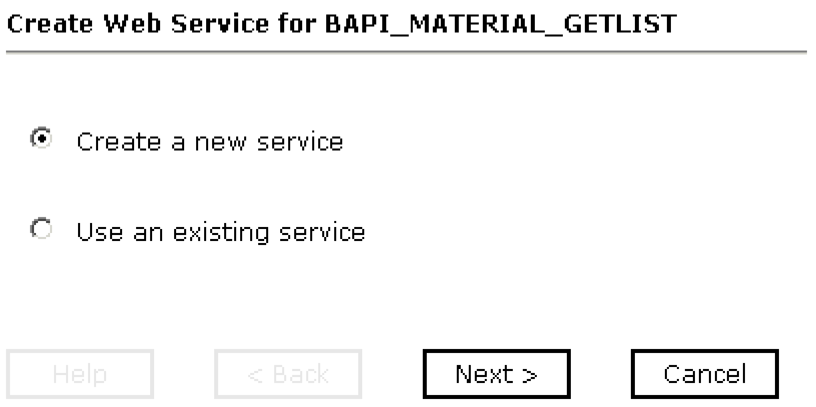


Properties fo

Property	
iwaf.descript	
BAPI	
Descriptio	
Business Object	Material
Business Method	GetList
Dialog	false
Static	true
Obsolete	false

3. Select *Create Integration Business Services.*

The Create Web Service for BAPI_MATERIAL_GETLIST pane opens on the right with options to create a new service or use an existing service.

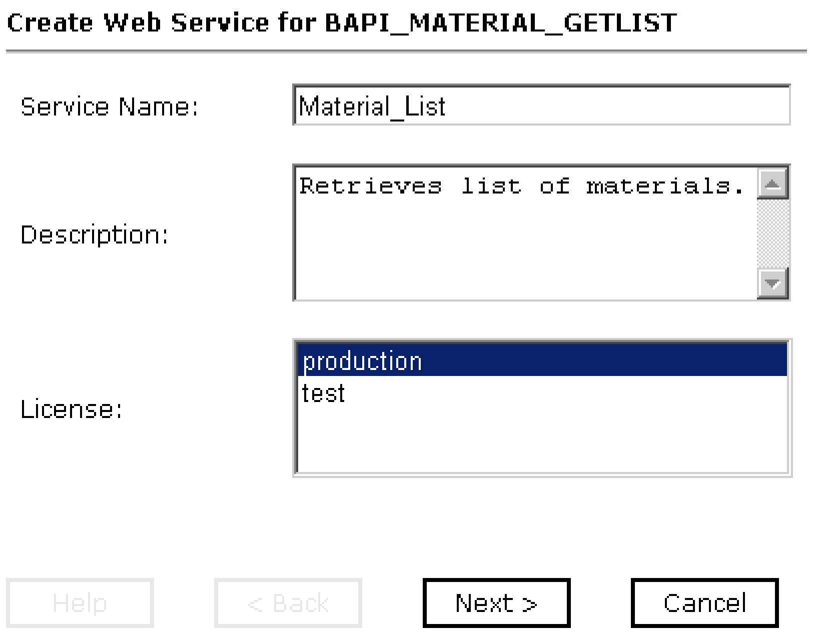

Create Web Service for BAPI_MATERIAL_GETLIST

[✓ Create a new service]

Use an existing service

4. Select *Create a new service* and click *Next*.

A Create Web Service pane opens where you provide the specific information for the Integration Business Service you are defining as shown in the following image.


Create Web Service for BAPI_MATERIAL_GETLIST

Service Name: Material_List

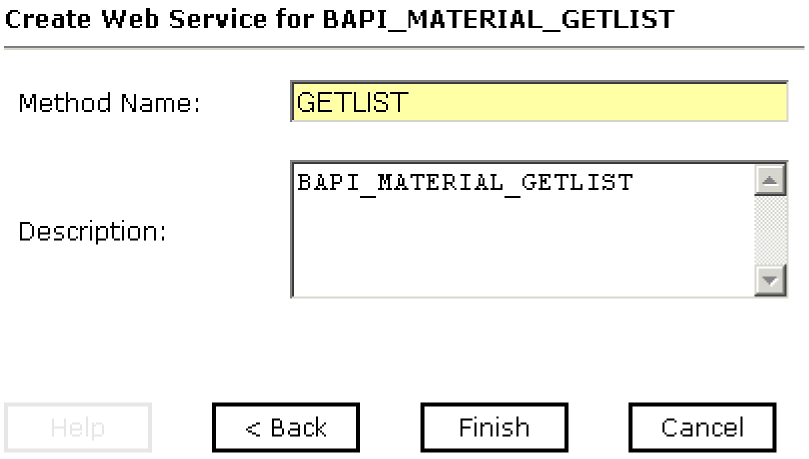
Description: Retrieves list of materials.

License: [✓ production] test

   a. In the Service Name field, type a name for the Integration Business Service.
   b. In the Description field, type a brief description (optional).
   c. In the License field, select the license definition you want to use.

5. Click *Next*.

   A second Create Web Service pane opens on the right that includes fields for the method name and a description as shown in the following image.



Create Web Service for BAPI_MATERIAL_GETLIST

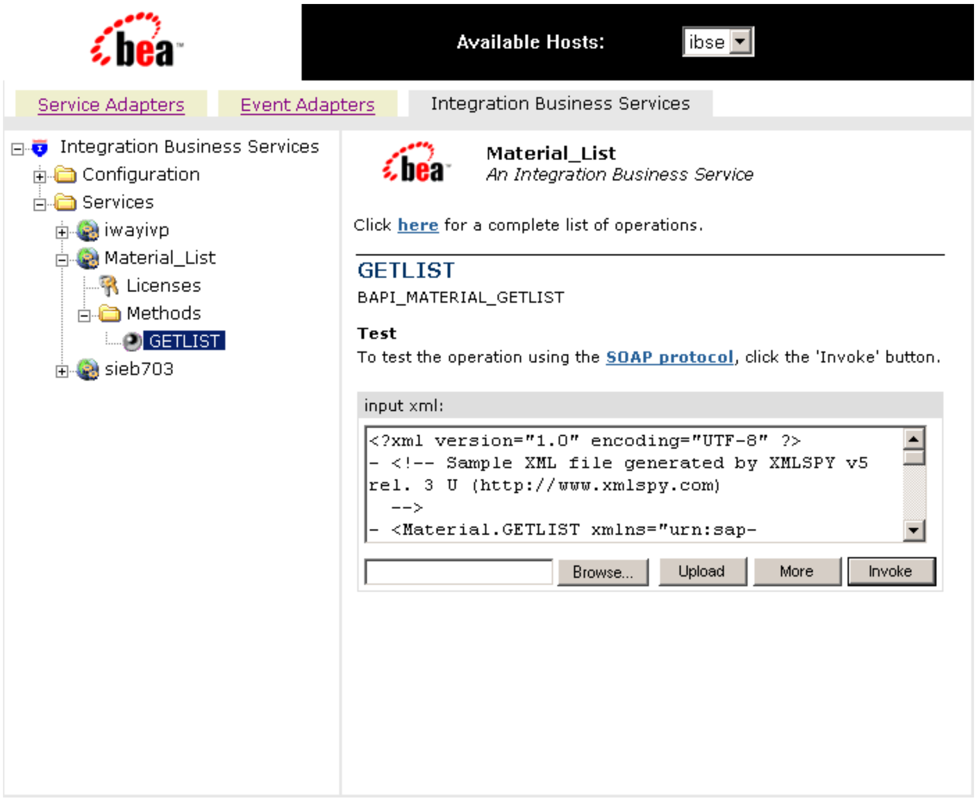
Method Name: GETLIST

Description: BAPI_MATERIAL_GETLIST

   a. In the Method Name field, type a descriptive name for the method.
   b. In the Description field, type a brief description for the method (optional).

6. Click *Finish*.

The Integration Business Services tab is active on the right as shown in the following image.



Material_List
An Integration Business Service

Click **here** for a complete list of operations.

GETLIST

BAPI_MATERIAL_GETLIST

Test

To test the operation using the **SOAP protocol**, click the 'Invoke' button.

input xml:

```
<?xml version="1.0" encoding="UTF-8" ?>
- <!-- Sample XML file generated by XMLSPY v5
rel. 3 U (http://www.xmlspy.com)
  -->
- <Material.GETLIST xmlns="urn:sap-
```

All of the available services that were created appear in the left pane. The Material_List service node is expanded, and the GETLIST method is automatically selected.

The test pane for the GETLIST method opens in the right pane.

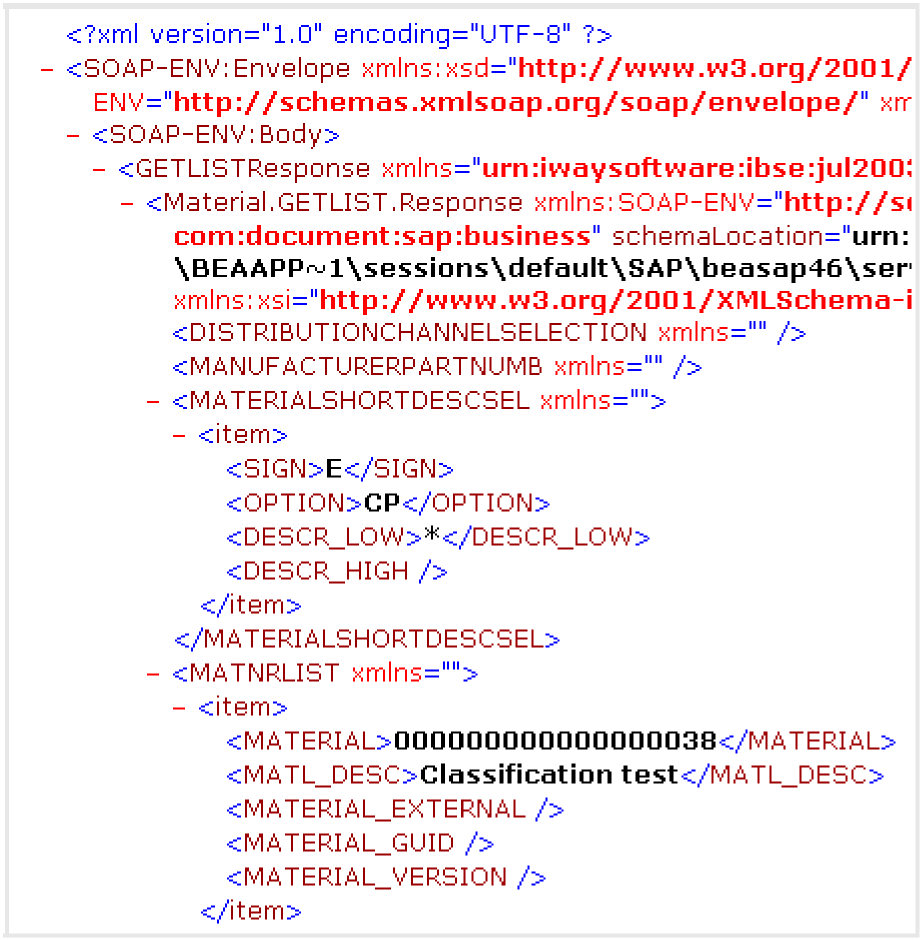
7. Enter an XML instance of the schema you generated previously for the SAP business component.

   The document queries the service in the input xml field.

   To use the identical sample input XML illustrated in this example, see *Sample Integration Business Services Input XML* on page 4-9.

8. Click *Invoke*.

The result appears in the right pane as shown in the following image.



```
  <?xml version="1.0" encoding="UTF-8" ?>
- <SOAP-ENV:Envelope xmlns:xsd="http://www.w3.org/2001/
    ENV="http://schemas.xmlsoap.org/soap/envelope/" xm
  - <SOAP-ENV:Body>
    - <GETLISTResponse xmlns="urn:iwaysoftware:ibse:jul200:
      - <Material.GETLIST.Response xmlns:SOAP-ENV="http://s
          com:document:sap:business" schemaLocation="urn:
          \BEAAPP~1\sessions\default\SAP\beasap46\ser
          xmlns:xsi="http://www.w3.org/2001/XMLSchema-i
          <DISTRIBUTIONCHANNELSELECTION xmlns="" />
          <MANUFACTURERPARTNUMB xmlns="" />
        - <MATERIALSHORTDESCSEL xmlns="">
          - <item>
              <SIGN>E</SIGN>
              <OPTION>CP</OPTION>
              <DESCR_LOW>*</DESCR_LOW>
              <DESCR_HIGH />
            </item>
          </MATERIALSHORTDESCSEL>
        - <MATNRLIST xmlns="">
          - <item>
              <MATERIAL>000000000000000038</MATERIAL>
              <MATL_DESC>Classification test</MATL_DESC>
              <MATERIAL_EXTERNAL />
              <MATERIAL_GUID />
              <MATERIAL_VERSION />
            </item>
```

Example: Sample Integration Business Services Input XML

The following input XML retrieves a list of materials using the SAP BAPI_MATERIAL_GETLIST method.

```
  <?xml version="1.0" encoding="UTF-8" ?>
- <!-- Sample XML file generated by XMLSPY v5 rel. 3 U
(http://www.xmlspy.com)
  -->
- <Material.GETLIST xmlns="urn:sap-com:document:sap:business"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="urn:sap-com:document:sap:business
C:\temp\service_BAPI_MATERIAL_GETLIST.xsd">
  <MAXROWS>1000</MAXROWS>
- <DISTRIBUTIONCHANNELSELECTION>
- <item>
  <SIGN />
  <OPTION />
  <DISTR_CHAN_LOW />
  <DISTR_CHAN_HIGH />
  </item>
  </DISTRIBUTIONCHANNELSELECTION>
- <MANUFACTURERPARTNUMB>
- <item>
  <MANU_MAT />
  <MFR_NO />
  </item>
  </MANUFACTURERPARTNUMB>
- <MATERIALSHORTDESCSEL>
- <item>
  <SIGN />
  <OPTION />
  <DESCR_LOW />
  <DESCR_HIGH />
  </item>
  </MATERIALSHORTDESCSEL>
- <MATNRLIST>
- <item>
  <MATERIAL />
  <MATL_DESC />
  <MATERIAL_EXTERNAL />
  <MATERIAL_GUID />
  <MATERIAL_VERSION />
  </item>
  </MATNRLIST>
- <MATNRSELECTION>
- <item>
  <SIGN>E</SIGN>
  <OPTION>BT</OPTION>
  <MATNR_LOW>1000</MATNR_LOW>
  <MATNR_HIGH>1010</MATNR_HIGH>
  </item>
  </MATNRSELECTION>
- <PLANTSELECTION>
- <item>
  <SIGN />
  <OPTION />
  <PLANT_LOW />
  <PLANT_HIGH />
  </item>
  </PLANTSELECTION>
- <RETURN>
- <item>
  <TYPE />
  <ID />
  <NUMBER />
  <MESSAGE />
  <LOG_NO />
  <LOG_MSG_NO />
  <MESSAGE_V1 />
  <MESSAGE_V2 />
  <MESSAGE_V3 />
  <MESSAGE_V4 />
  <PARAMETER />
  <ROW>0</ROW>
  <FIELD />
  <SYSTEM />
  </item>
  </RETURN>
- <SALESORGANISATIONSELECTION>
- <item>
  <SIGN />
  <OPTION />
  <SALESORG_LOW />
  <SALESORG_HIGH />
  </item>
  </SALESORGANISATIONSELECTION>
- <STORAGELOCATIONSELECT>
- <item>
  <SIGN />
  <OPTION />
  <STLOC_LOW />
  <STLOC_HIGH />
  </item>
  </STORAGELOCATIONSELECT>
  </Material.GETLIST>
```

Generating WSDL From a Web Service

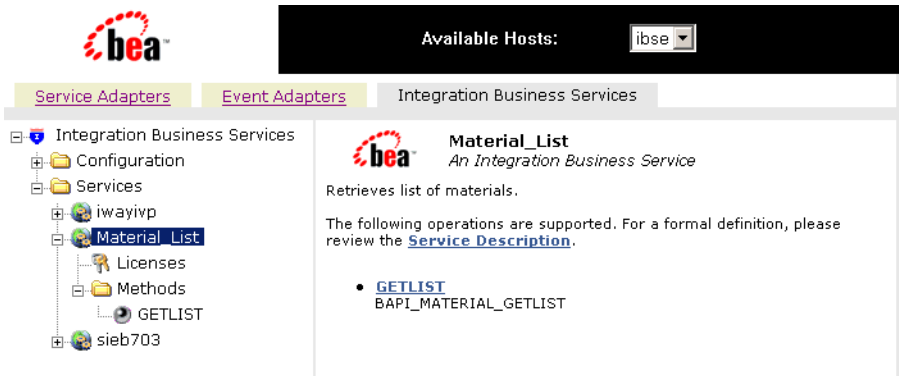
Generating WSDL (Web Services Description Language) from a Web service enables you to make the Web service available to other services within a host server such as the BEA WebLogic Server.

Procedure: How to Generate WSDL From a Web Service

To generate WSDL from a Web service:

1. Click the *Integration Business Services* tab.
2. To view the service for which you want to generate WSDL, in the left pane, expand the list of services.
3. Select the service, for example, Material_List.

   The following image shows Material_List service selected in the left pane. The link for the service appears in the right pane.



Material_List
An Integration Business Service

Retrieves list of materials.

The following operations are supported. For a formal definition, please review the **Service Description**.

- **GETLIST**
  BAPI_MATERIAL_GETLIST

   a. Right-click the *Service Description* link.

   b. Select *Save Target As*.

4. Choose a location for the file and add a .wsdl file extension.
5. Click *Save*.

   For example, saving a Web service called Material_List for an SAP R/3 creates a file named Material_List.wsdl.

   Note: The file extension must be .wsdl.

The following is an example of a WSDL file for a Web service called BAPI_MATERIAL_GET_DETAIL.

```
<definitions xmlns:rfc="urn:iwaysoftware:ibse:jul2003:BAPI:response"
 xmlns:tns="urn:schemas-iwaysoftware-com:iwse"
 targetNamespace="urn:schemas-iwaysoftware-com:iwse"
 xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
 xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
 xmlns:m11="urn:iwaysoftware:ibse:jul2003:BAPI:response"
 xmlns:tm="http://microsoft.com/wsdl/mime/textMatching/"
 xmlns="http://schemas.xmlsoap.org/wsdl/"
 xmlns:xs="http://www.w3.org/2001/XMLSchema"
 xmlns:m1="urn:iwaysoftware:ibse:jul2003:BAPI"
 xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/">
 <types>
  <xs:schema targetNamespace="urn:schemas-iwaysoftware-com:iwse"
   elementFormDefault="qualified">
     <xs:element name="ibsinfo">
        <xs:complexType>
          <xs:sequence>
           <xs:element type="xs:string" name="service"/>
           <xs:element type="xs:string" name="method"/>
           <xs:element type="xs:string" name="license"/>
           <xs:element type="xs:string" minOccurs="0"
            name="disposition"/>
           <xs:element type="xs:string" minOccurs="0" name="Username"/>
           <xs:element type="xs:string" minOccurs="0" name="Password"/>
           <xs:element type="xs:string" minOccurs="0" name="language"/>
         </xs:sequence>
        </xs:complexType>
     </xs:element>
   </xs:schema>
  <xs:schema targetNamespace="urn:schemas-iwaysoftware-com:iwse"
   elementFormDefault="qualified">
    <xs:element name="adapterexception">
      <xs:complexType>
        <xs:sequence>
          <xs:element type="xs:string" name="error"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
  </xs:schema>
  <xs:schema xmlns:rfc="urn:iwaysoftware:ibse:jul2003:BAPI"
   targetNamespace="urn:iwaysoftware:ibse:jul2003:BAPI"
   xmlns:m1="urn:iwaysoftware:ibse:jul2003:BAPI"
   elementFormDefault="qualified">
     <xs:element name="BAPI">
       <xs:complexType>
         <xs:sequence>
           <xs:element name="BAPI_MATERIAL_GET_DETAIL">
             <xs:complexType>
               <xs:all>
                 <xs:element minOccurs="1" name="MATERIAL">
                   <xs:simpleType>
                     <xs:restriction base="xs:string">
                       <xs:maxLength value="18"/>
                     </xs:restriction>
                   </xs:simpleType>
                 </xs:element>
                <xs:element minOccurs="0" name="PLANT">
                  <xs:simpleType>
                    <xs:restriction base="xs:string">
                      <xs:maxLength value="4"/>
                    </xs:restriction>
                  </xs:simpleType>
                </xs:element>
                <xs:\element minOccurs="0" name="VALUATIONAREA">
                  <xs:simpleType>
                    <xs:restriction base="xs:string">
                      <xs:maxLength value="4"/>
                    </xs:restriction>
                  </xs:simpleType>
                </xs:element>
                <xs:element minOccurs="0" name="VALUATIONTYPE">
                  <xs:simpleType>
                   <xs:restriction base="xs:string">
                      <xs:maxLength value="10"/>
                    </xs:restriction>
                  </xs:simpleType>
                </xs:element>
              </xs:all>
            </xs:complexType>
          </xs:element>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
  </xs:schema>
  <xs:schema xmlns:rfc="urn:iwaysoftware:ibse:jul2003:BAPI:response"
   targetNamespace="urn:iwaysoftware:ibse:jul2003:BAPI:response"
   xmlns:m11="urn:iwaysoftware:ibse:jul2003:BAPI:response"
   elementFormDefault="qualified">
     <xs:element name="BAPIResponse">
       <xs:complexType>
         <xs:sequence>
           <xs:element name="BAPI_MATERIAL_GET_DETAIL.Response">
              <xs:complexType>
            <xs:all>
              <xs:element type="rfc:BAPIMATDOC" minOccurs="0"
               name="MATERIALPLANTDATA"/>
              <xs:element type="rfc:BAPIMATDOBEW" minOccurs="0"
               name="MATERIALVALUATIONDATA"/>
              <xs:element type="rfc:BAPIMATDOA" minOccurs="0"
               name="MATERIAL_GENERAL_DATA"/>
              <xs:element type="rfc:BAPIRETURN" minOccurs="0"
               name="RETURN"/>
            </xs:all>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
      <xs:attribute type="xs:string" use="required" name="cid"/>
    </xs:complexType>
  </xs:element>
  <xs:complexType name="BAPIMATDOC">
    <xs:sequence>
      <xs:element name="PUR_GROUP">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:maxLength value="3"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="ISSUE_UNIT">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:maxLength value="3"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="BAPIMATDOBEW">
    <xs:sequence>
      <xs:element name="PRICE_CTRL">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:maxLength value="1"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="MOVING_PR">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:maxLength value="23"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="STD_PRICE">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:maxLength value="23"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="PRICE_UNIT">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:maxLength value="5"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="CURRENCY">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:maxLength value="5"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="CURRENCY_ISO">
        <xs:simpleType>
           <xs:restriction base="xs:string">
             <xs:maxLength value="3"/>
           </xs:restriction>
        </xs:simpleType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="BAPIMATDOA">
    <xs:sequence>
      <xs:element name="MATL_DESC">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:maxLength value="40"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="OLD_MAT_NO">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:maxLength value="18"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
```

```
<xs:element name="MATL_TYPE">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:maxLength value="4"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name="IND_SECTOR">
   <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:maxLength value="1"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name="DIVISION">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:maxLength value="2"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name="MATL_GROUP">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:maxLength value="9"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name="PROD_HIER">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:maxLength value="18"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name="BASIC_MATL">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:maxLength value="14"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name="STD_DESCR">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:maxLength value="18"/>
    </xs:restriction>
  </xs:simpleType>
```

```
</xs:element>
<xs:element name="LAB_DESIGN">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:maxLength value="3"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name="PROD_MEMO">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:maxLength value="18"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name="PAGEFORMAT">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:maxLength value="4"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name="CONTAINER">
<xs:simpleType>
  <xs:restriction base="xs:string">
    <xs:maxLength value="2"/>
  </xs:restriction>
</xs:simpleType>
</xs:element>
<xs:element name="STOR_CONDS">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:maxLength value="2"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name="TEMP_CONDS">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:maxLength value="2"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name="BASE_UOM">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:maxLength value="3"/>
    </xs:restriction>
    </xs:simpleType>
  </xs:element>
  <xs:element name="EAN_UPC">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:maxLength value="18"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:element>
  <xs:element name="EAN_CAT">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:maxLength value="2"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:element>
  <xs:element name="SIZE_DIM">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:maxLength value="32"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name="GROSS_WT">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:maxLength value="13"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name="NET_WEIGHT">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:maxLength value="13"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name="UNIT_OF_WT">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:maxLength value="3"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name="VOLUME">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:maxLength value="13"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name="VOLUMEUNIT">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:maxLength value="3"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name="LENGTH">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:maxLength value="13"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name="WIDTH">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:maxLength value="13"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name="HEIGHT">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:maxLength value="13"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name="UNIT_DIM">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:maxLength value="3"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name="MANU_MAT">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:maxLength value="40"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name="MFR_NO">
   <xs:simpleType>
     <xs:restriction base="xs:string">
```

```
        <xs:maxLength value="10"/>
     </xs:restriction>
   </xs:simpleType>
</xs:element>
<xs:element name="BASE_UOM_ISO">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:maxLength value="3"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name="UNIT_OF_WT_ISO">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:maxLength value="3"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name="VOLUMEUNIT_ISO">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:maxLength value="3"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name="UNIT_DIM_ISO">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:maxLength value="3"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name="CREATED_ON">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:maxLength value="8"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name="CREATED_BY">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:maxLength value="12"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name="LAST_CHNGE">
  <xs:simpleType>
```

```
            <xs:restriction base="xs:string">
              <xs:maxLength value="8"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:element>
        <xs:element name="CHANGED_BY">
          <xs:simpleType>
            <xs:restriction base="xs:string">
              <xs:maxLength value="12"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:element>
        <xs:element name="MATL_CAT">
          <xs:simpleType>
            <xs:restriction base="xs:string">
              <xs:maxLength value="2"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:element>
        <xs:element name="EMPTIESBOM">
          <xs:simpleType>
            <xs:restriction base="xs:string">
              <xs:maxLength value="1"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:element>
        <xs:element name="BASIC_MATL_NEW">
          <xs:simpleType>
            <xs:restriction base="xs:string">
              <xs:maxLength value="48"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
    <xs:complexType name="BAPIRETURN">
      <xs:sequence>
        <xs:element name="TYPE">
          <xs:simpleType>
            <xs:restriction base="xs:string">
              <xs:maxLength value="1"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:element>
        <xs:element name="CODE">
          <xs:simpleType>
            <xs:restriction base="xs:string">
              <xs:maxLength value="5"/>
```

```
     </xs:restriction>
   </xs:simpleType>
 </xs:element>
 <xs:element name="MESSAGE">
   <xs:simpleType>
     <xs:restriction base="xs:string">
       <xs:maxLength value="220"/>
     </xs:restriction>
   </xs:simpleType>
 </xs:element>
 <xs:element name="LOG_NO">
   <xs:simpleType>
     <xs:restriction base="xs:string">
       <xs:maxLength value="20"/>
     </xs:restriction>
   </xs:simpleType>
 </xs:element>
 <xs:element name="LOG_MSG_NO">
   <xs:simpleType>
     <xs:restriction base="xs:string">
       <xs:maxLength value="6"/>
     </xs:restriction>
   </xs:simpleType>
 </xs:element>
 <xs:element name="MESSAGE_V1">
   <xs:simpleType>
     <xs:restriction base="xs:string">
       <xs:maxLength value="50"/>
     </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name="MESSAGE_V2">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:maxLength value="50"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name="MESSAGE_V3">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:maxLength value="50"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name="MESSAGE_V4">
  <xs:simpleType>
    <xs:restriction base="xs:string">
```

```
          <xs:maxLength value="50"/>
         </xs:restriction>
       </xs:simpleType>
     </xs:element>
   </xs:sequence>
  </xs:complexType>
 </xs:schema>
</types>
<message name="BAPIIn"><part element="m1:BAPI" name="parameters"/>
</message>
<message name="BAPIOut"><part element="m11:BAPIResponse"
 name="parameters"/>
</message>
<message name="BAPI_MATERIAL_GET_DETAILHeader">
 <part element="tns:ibsinfo" name="header"/>
</message>
<message name="AdapterException"><part element="tns:adapterexception"
 name="fault"/>
</message>
<portType name="BAPI_MATERIAL_GET_DETAILSoap">
 <operation name="BAPI">
  <documentation/>
   <input message="tns:BAPIIn"/>
   <output message="tns:BAPIOut"/>
   <fault message="tns:AdapterException" name="AdapterExceptionFault"/>
  </operation>
</portType>
<binding type="tns:BAPI_MATERIAL_GET_DETAILSoap"
 name="BAPI_MATERIAL_GET_DETAILSoap">
  <soap:binding style="document"
   transport="http://schemas.xmlsoap.org/soap/http"/>
  <operation name="BAPI">
    <soap:operation style="document"
     soapAction="BAPI_MATERIAL_GET_DETAIL.BAPIRequest@test@@"/>
    <input>
      <soap:body use="literal"/>
      <soap:header part="header"
       message="tns:BAPI_MATERIAL_GET_DETAILHeader" use="literal"/>
    </input>
    <output>
      <soap:body use="literal"/>
    </output>
    <fault name="AdapterExceptionFault">
      <soap:fault use="literal" name="AdapterExceptionFault"/>
    </fault>
  </operation>
</binding>
<service name="BAPI_MATERIAL_GET_DETAIL">
```

```
    <documentation>BAPI_MATERIAL_GET_DETAIL
    </documentation>
    <port binding="tns:BAPI_MATERIAL_GET_DETAILSoap"
     name="BAPI_MATERIAL_GET_DETAILSoap1">
      <soap:address
       location="http://GERBER-2K.ibi.com:7001/ibse/IBSEServlet
       /XDSOAPRouter"/>
    </port>
  </service>
</definitions>
```

Identity Propagation

If you test or execute a Web service using a third-party XML editor, for example, XMLSPY, the user name and password values that you specify in the SOAP header must be valid. The values are used to connect to SAP. The user name and password values that you provided for SAP when you created a target using Application Explorer are overwritten for this Web service request.

The following is a sample SOAP header that is included in the WSDL file for a Web service:

```
<SOAP-ENV:Header>
    <m:ibsinfo xmlns:m="urn:schemas-iwaysoftware-com:iwse">
        <m:service>String</m:service>
        <m:method>String</m:method>
        <m:license>String</m:license>
        <m:disposition>String</m:disposition>
        <m:Username>String</m:Username>
        <m:Password>String</m:Password>
        <m:language>String</m:language>
    </m:ibsinfo>
</SOAP-ENV:Header>
```

Note: You can remove the following tags from the SOAP header, since they are not required:

```
<m:disposition>String</m:disposition>
```

```
<m:language>String</m:language>
```

CHAPTER 5

Configuring the Event Adapter for SAP

Topics:

- Understanding Event Functionality
- Creating, Editing, or Deleting a Port
- Creating, Editing, or Deleting a Channel
- Synchronous Event Processing

This section describes how to use Application Explorer to connect to SAP and listen for events.

The functionality of the Application Explorer is standard despite the deployment type. This section uses the Java™ servlet implementation of Application Explorer to provide graphic examples.

For information on running Application Explorer in WebLogic Workshop, see Appendix B, *Using Application Explorer in BEA WebLogic Workshop for Event Handling*.

Understanding Event Functionality

Events are generated as a result of activity on an application system. You can use events to trigger an action in your application.

Applications or functions within SAP may broadcast processing information at predefined points, called events. You must configure an event listener if you are to receive events from SAP. For example, the SAP Business Object, Material, may raise the event status Material, assigned when a material is created. If you wish to consume this event, you must configure an event listener to capture this event within SAP and transmit the event notification to your system.

To create an event, you must create a port and a channel using Application Explorer.

The following is a description of how ports and channels work.

- Port

  A port associates a particular business object exposed by an adapter with a particular disposition. A disposition defines the protocol and location of the event data. The port defines the end point of the event consumption. For more information, see *Creating, Editing, or Deleting a Port on page 5-2*.

- Channel

  A channel represents configured connections to particular instances of back-end or other types of systems. A channel binds one or more event ports to a particular listener managed by an adapter. For more information, see *Creating, Editing, or Deleting a Channel* on page 5-16.

Creating, Editing, or Deleting a Port

The following procedures describe how to create, edit, or delete an event port using Servlet Application Explorer. You can create a port for an SAP business function from the Service Adapters tab or from the Event Adapters tab.

When you use Application Explorer with an Integration Business Services Engine (iBSE) implementation, the following port dispositions are available:

- File
- iBSE
- MSMQ
- JMS queue
- SOAP
- HTTP

- MQ Series
- MAIL

Note: The MAIL disposition option will be supported in a future release.

The following dispositions are available when using Application Explorer in conjunction with a JCA connector implementation:

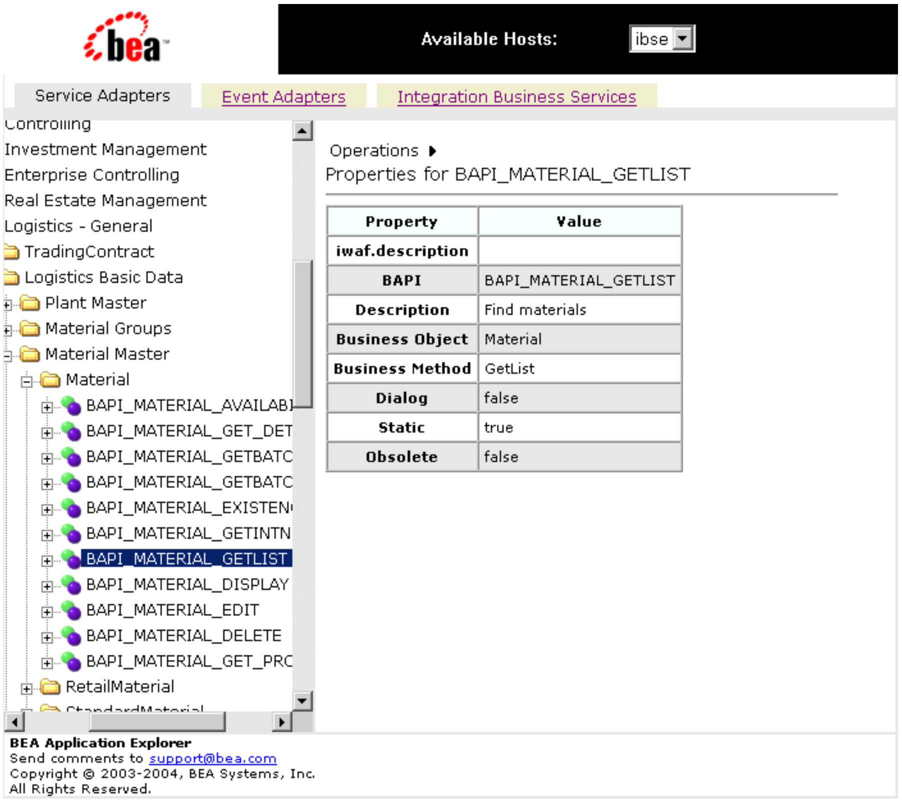
- File
- HTTP
- JMS Queue
- MQ Series

Procedure: How to Create a Port for the File Disposition

To create a port for the File disposition using Application Explorer:

1. Click the *Service Adapters* tab.
2. From the Business Object Repository, select the *BAPI_MATERIAL_GETLIST* method.

The following image shows the Operations menu in the right pane and a table listing properties and values for the BAPI method named BAPI_MATERIAL_GETLIST.



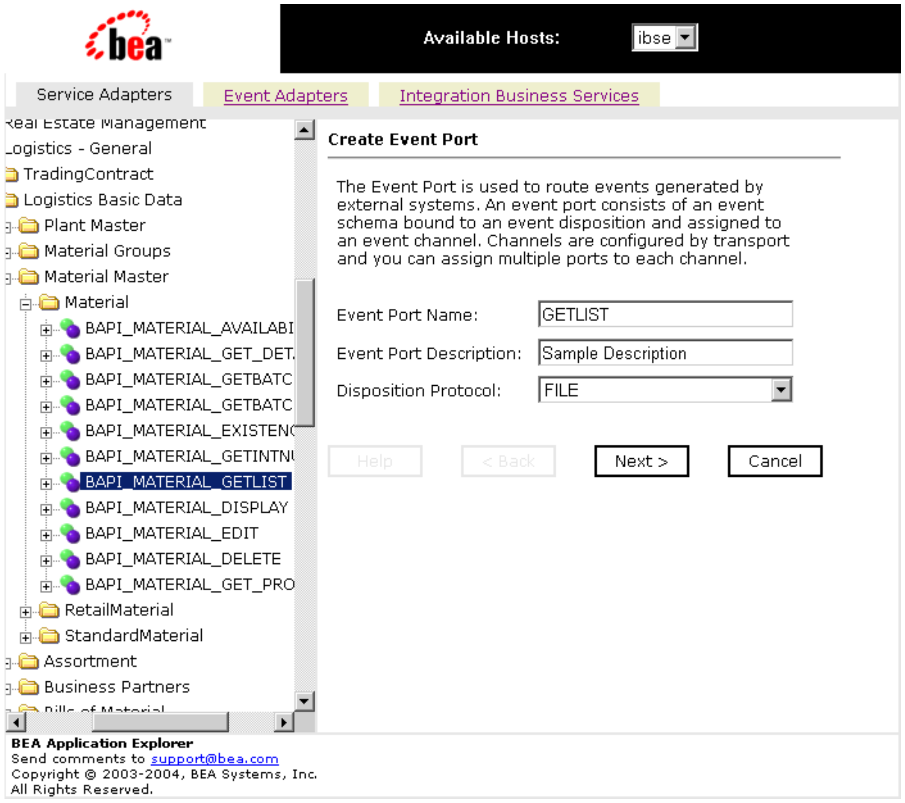
Properties for BAPI_MATERIAL_GETLIST

Property	Value
iwaf.description	
BAPI	BAPI_MATERIAL_GETLIST
Description	Find materials
Business Object	Material
Business Method	GetList
Dialog	false
Static	true
Obsolete	false

BEA Application Explorer
Send comments to support@bea.com
Copyright © 2003-2004, BEA Systems, Inc.
All Rights Reserved.

3. In the right pane, move the pointer over *Operations* and select *Create Event Port*.

The Create Event Port pane opens on the right as shown in the following image.



Create Event Port

The Event Port is used to route events generated by external systems. An event port consists of an event schema bound to an event disposition and assigned to an event channel. Channels are configured by transport and you can assign multiple ports to each channel.

Event Port Name:	GETLIST
Event Port Description:	Sample Description
Disposition Protocol:	FILE

BEA Application Explorer
Send comments to support@bea.com
Copyright © 2003-2004, BEA Systems, Inc.
All Rights Reserved.

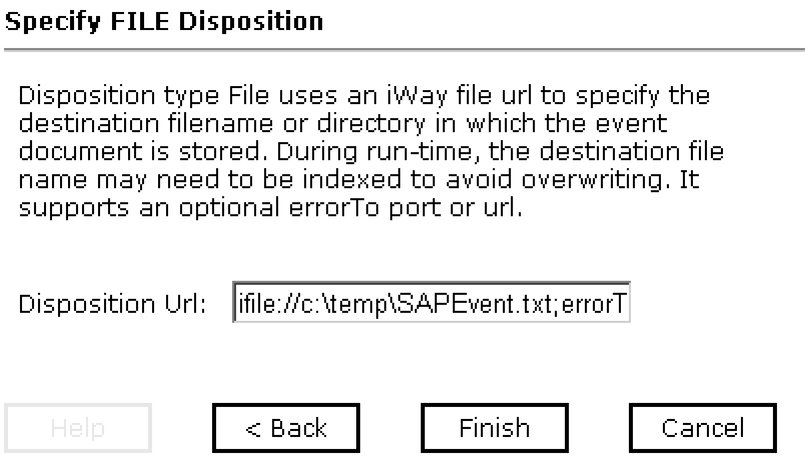
a. In the Event Port Name field, type a name.

b. In the Event Port Description field, type a brief description (optional).

c. From the Disposition Protocol drop-down list, select *FILE*.

4. Click *Next*.

The Specify FILE Disposition pane opens on the right as shown in the following image and includes information about the File disposition.



Specify FILE Disposition

Disposition type File uses an iWay file url to specify the destination filename or directory in which the event document is stored. During run-time, the destination file name may need to be indexed to avoid overwriting. It supports an optional errorTo port or url.

Disposition Url: ifile://c:\temp\SAPEvent.txt;errorT

5. In the Disposition Url field, type a destination where the event data is written.

   When pointing Application Explorer to an **iBSE** deployment, specify the destination file using the following format:

   ```
   ifile://location;[errorTo=errorDest]
   ```

   When pointing Application Explorer to a **JCA** deployment, specify the destination file using the following format:

   ```
   location
   ```

   The following table lists and describes the disposition parameters for File.

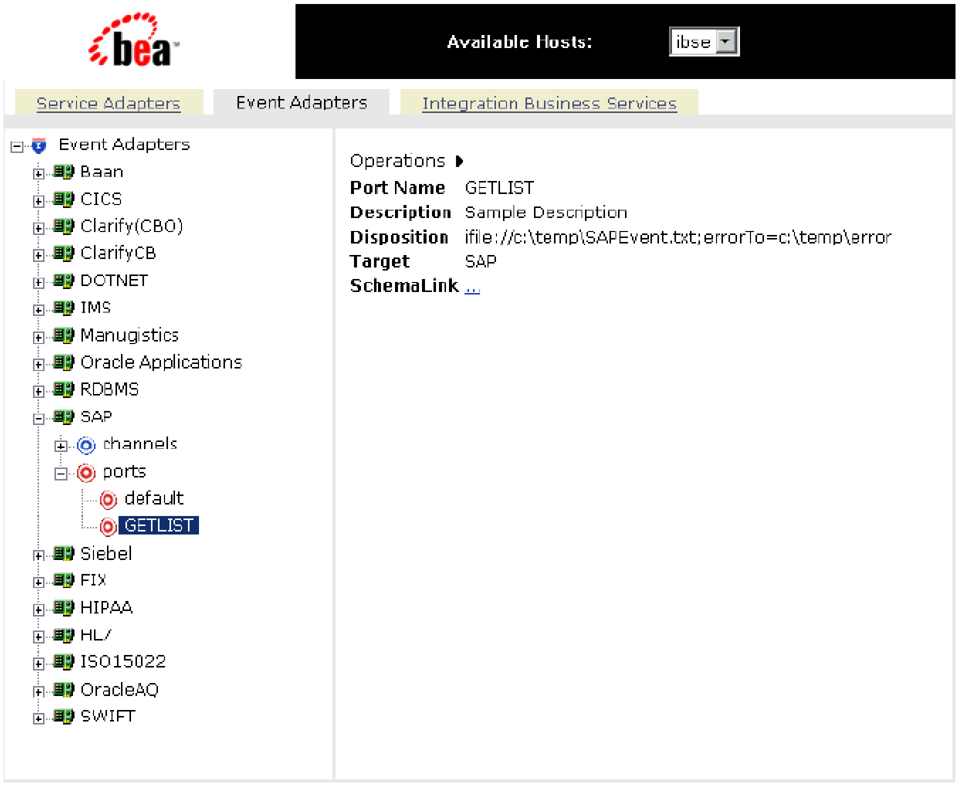
Parameter	**Description**
location	Full directory path and file name to which the data is written.
errorDest	Location to which error logs are sent. Optional. Predefined port name or another disposition URL. The URL must be complete, including the protocol.

   For example:

   ```
   ifile://c:\temp\SAPEvent.txt;errorTo=ifile://c:\temp\error
   ```

6. Click *Finish*.

The Event Adapters tab becomes available. The event port appears under the ports node in the left pane. In the right pane, a summary of the information associated with the port appears.



Port Name GETLIST
Description Sample Description
Disposition ifile://c:\temp\SAPEvent.txt;errorTo=c:\temp\error
Target SAP
SchemaLink ...

7. To view the event schema that was created for the event port, click *SchemaLink*.

   You are now ready to associate the event port for File with a channel. For more information, see *Creating, Editing, or Deleting a Channel* on page 5-16.

Procedure: How to Create a Port for the iBSE Disposition

The iBSE disposition enables an event to launch an Integration Business Services method. To create a port for an iBSE disposition using Application Explorer:

1. Click the *Event Adapters* tab.
2. In the left pane, expand the *SAP* node.
3. Select the *ports* node.
4. Move the pointer over *Operations* and select *Add a new port*.

The Create New Port pane opens on the right.

a. In the Name field, type a name.

b. In the Description field, type a brief description (optional).

c. From the Disposition Protocol drop-down list, select *IBSE*.

d. In the Disposition field, enter an iBSE destination using the following format:

```
ibse:svcName.mthName;[responseTo=responseTo];[errorTo=errorDest]
```

The following table lists and describes the disposition parameters for iBSE.

Parameter	**Description**
svcName	Name of the service created with iBSE.
mthName	Name of the method created for the Web service.
responseTo	Location to which responses to the Web service are posted. Optional. Predefined port name or another disposition URL. The URL must be complete, including the protocol.
errorDest	Location to which error logs are sent. Optional. Predefined port name or another disposition URL. The URL must be complete, including the protocol.

5. Click *OK*.

The port appears under the ports node in the left pane. In the right pane, a summary of the information associated with the port appears.

Procedure: How to Create a Port for the MSMQ Disposition

The MSMQ disposition supports public and private queues. To create a port for an MSMQ disposition using Application Explorer:

1. Click the *Event Adapters* tab.

2. In the left pane, expand the *SAP* node.

3. Select the *ports* node.

4. Move the pointer over *Operations* and select *Add a new port*.

The Create New Port pane opens on the right.

a. In the Name field, type a name.

b. In the Description field, type a brief description (optional).

c. From the Disposition Protocol drop-down list, select *MSMQ*.

d. In the Disposition field, enter an MSMQ destination in the format:

```
msmq://host/private$/qName;[errorTo=errorDest]
```

The following table lists and describes the dispostion parameters for MSMQ.

Parameter	Description
host	Name of the host on which the Microsoft Queuing system runs.
queueType	Type of queue. For private queues, enter *Private$*. Private queues are queues that are not published in Active Directory. They appear only on the local computer that contains them. Private queues are accessible only by Message Queuing applications that recognize the full path name or format name of the queue.
qName	Name of the queue in which messages are placed.
errorTo	Location where error documents are sent. Predefined port name or another full URL. Optional.

5. Click *OK*.

The port appears under the ports node in the left pane. In the right pane, a summary of the information associated with the port appears.

Procedure: How to Create a Port for the JMS Queue Disposition

The JMS queue disposition enables an event to be enqueued to a JMS queue. To create a port for a JMS queue disposition using Application Explorer:

1. Click the *Event Adapters* tab.

2. In the left pane, expand the *SAP* node.

3. Select the *ports* node.

4. Move the pointer over *Operations* and select *Add a new port*.

The Create New Port window opens in the right pane.

a. In the Name field, type a name.

b. In the Description field, type a brief description (optional).

c. From the Disposition Protocol drop-down list, select *JMSQ*.

d. In the Disposition field, enter a JMS destination.

When pointing Application Explorer to an **iBSE** deployment, use the following format:

```
zjmsq:myQueueName@myQueueFac;jndiurl=[myurl];
jndifactory=[myfactory];user=[user];password=[xxx];
errorTo=[pre-defined port name or another disposition url]
```

When pointing Application Explorer to a **JCA** deployment, use the following format:

```
jms:jmsqueue@jmsfactory;jndiurl=;jndifactory=;
```

The following table lists and describes the parameters for the disposition.

Parameter	**Description**
queue	JNDI name of a queue to which events are emitted.
Connection Factory	A resource that contains information about the JMS Server. The WebLogic connection factory is: `javax.jms.QueueConnectionFactory`
jndiurl	The URL to use to contact the JNDI provider. The syntax of this URL depends on which JNDI provider is being used. This value corresponds to the standard JNDI property, `java.naming.provider.url` For BEA WebLogic Server this is: `t3://`*`host:port`* where: *`host`* Is the machine name where WebLogic Server is installed. *`port`* Is the port on which WebLogic server is listening. The default port if not changed at installation is 7001.

Parameter	Description
jndifactory	Is JNDI context.INITIAL_CONTEXT_FACTORY and is provided by the JNDI service provider. For WebLogic Server, the WebLogic factory is: `weblogic.jndi.WLInitialContextFactory`
user	Valid user name required to access a JMS server.
password	Valid password required to access a JMS server.
errorTo	Location where error documents are sent. A predefined port name or another full URL. Optional.

5. Click *OK*.

The port appears under the ports node in the left pane. In the right pane, a summary of the information associated with the port appears.

Procedure: How to Create a Port for the SOAP Disposition

To create a port for a SOAP disposition using Application Explorer:

1. Click the *Event Adapters* tab.
2. In the left pane, expand the *SAP* node.
3. Select the *ports* node.
4. Move the pointer over *Operations* and select *Add a new port*.

   The Create New Port pane opens on the right.

   a. In the Name field, type a name.

   b. In the Description field, type a brief description (optional).

   c. From the Disposition Protocol drop-down list, select *SOAP*.

   d. In the Disposition field, enter a SOAP destination, using the following format:

   ```
   soap:[wsdl-url];soapaction=[myaction];
   method=[web service method];namespace=[namespace];
   responseTo=[pre-defined port name or another disposition URL];
   errorTo=[pre-defined port name or another disposition url]
   ```

The following table lists and describes the parameters for the disposition.

Parameter	**Description**
wsdl-url	The URL to the WSDL file that is required to create the SOAP message, for example: `http://localhost:7001/ibse/IBSEServlet/test/`*`webservice`*`.ibs?wsdl` where: *`webservice`* Is the name of the Web service you created using Application Explorer. To find this value, you can navigate to the Integration Business Services tab and open the Service Description link in a new window. The WSDL URL appears in the Address field. Alternatively, you can open the WSDL file in a third-party XML editor (for example, XMLSPY) and view the SOAP request settings.
soapaction	Method that is called by the SOAP disposition. This value can be found in the WSDL file.
method	Web service method you are using. Value is found in the WSDL file.
namespace	XML namespace you are using. Value is found in the WSDL file.
responseTo	Location to which responses are posted. Can be a predefined port name or another URL. Optional. The URL must be complete, including the protocol.
errorTo	Location where error documents are sent. A predefined port name or another full URL. Optional.

Note: To use the SOAP disposition with a synchronous event, use Remote Function Modules to generate the schema and WSDL file instead of using the Business Object Repository for any RFC or BAPI.

5. Click *OK*.

The event port appears under the ports node in the left pane. In the right pane, a summary of the information associated with the port appears.

Procedure: How to Create a Port for the HTTP Disposition

The HTTP disposition uses an HTTP URL to specify an HTTP end point to which the event document is posted. To create a port for an HTTP disposition using Application Explorer:

1. Click the *Event Adapters* tab.
2. In the left pane, expand the *SAP* node.
3. Select the *ports* node.
4. Move the pointer over *Operations* and select *Add a new port*.

   The Create New Port pane opens on the right.

   a. In the Name field, type a name.

   b. In the Description field, type a brief description (optional).

   c. From the Disposition Protocol drop-down list, select *HTTP*.

   d. In the Disposition field, enter an HTTP destination.

      When pointing Application Explorer to an **iBSE** deployment, specify the destination file using the following format:

      ```
      ihttp://url;responseTo=respDest
      ```

      When pointing Application Explorer to a **JCA** deployment, specify the destination file using the following format:

      ```
      http://host:port/uri
      ```

      The following table lists and describes the disposition parameters for HTTP.

Parameter	Description
url	The URL target for the post operation.
respDest	Location to which responses are posted. A predefined port name or another full URL. Optional. The URL must be complete, including the protocol.
host	Name of the host on which the Web server resides.
port	Port number on which the Web server is listening.
uri	Universal resource identifier that completes the URL specification.

5. Click *OK*.

The event port appears under the ports node in the left pane. In the right pane, a summary of the information associated with the port appears.

Procedure: How to Create a Port for the MQ Series Disposition

The MQSeries disposition enables an event to be enqueued to an MQSeries queue. Both queue manager and queue name may be specified. To create a port for an MQSeries disposition using Application Explorer:

1. Click the *Event Adapters* tab.
2. In the left pane, expand the *SAP* node.
3. Select the *ports* node.
4. Move the pointer over *Operations* and select *Add a new port*.

   The Create New Port pane opens on the right.

   a. In the Name field, type a name.

   b. In the Description field, type a brief description (optional).

   c. From the Disposition Protocol drop-down list, select *MQSeries*.

   d. In the Disposition field, enter an MQSeries destination.

   When pointing Application Explorer to an **iBSE** deployment, specify the destination file using the following format:

   ```
   mqseries:/qManager/qName;host=[hostname];
   port=[port];channel=[channnelname];
   errorTo=[pre-defined port name or another disposition url]
   ```

   When pointing Application Explorer to a **JCA** deployment, specify the destination file using the following format:

   ```
   mq:qmanager@respqueue;host=;port=;channel=
   ```

   The following table lists and describes the disposition parameters for MQSeries.

Parameter	**Description**
qManager	Name of the queue manager to which the server must connect.
qName or respqueue	Name of the queue where messages are placed.
host	Host on which the MQ Server is located (MQ Client only).

Parameter	Description
port	Number to connect to an MQ Server queue manager (MQ client only).
channel	Case-sensitive name of the channel that connects with the remote MQ Server queue manager (MQ client only). The default channel name for MQSeries is SYSTEM.DEF.SVRCONN.
errorTo	Location where error documents are sent. This can be a predefined port name or another full URL. Optional.

5. Click *OK*.

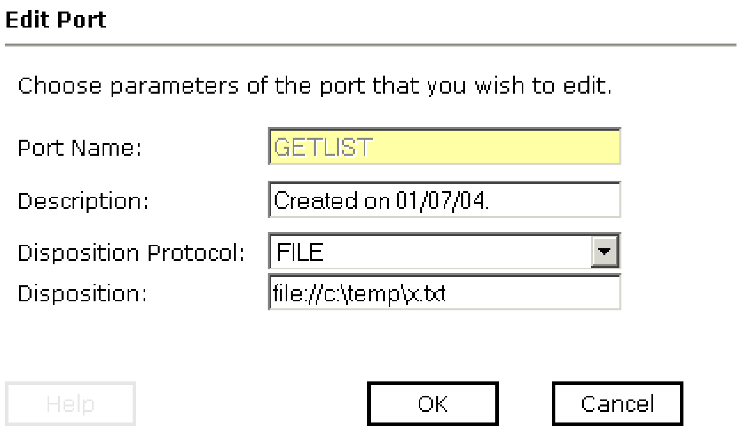
   The port appears under the ports node in the left pane. In the right pane, a summary of the information associated with the port appears.

Procedure: How to Edit an Event Port

To edit an event port:

1. Select the event port you want to edit.
2. In the right pane, move the pointer over *Operations* and select *Edit*.

   The Edit Port pane opens on the right, with fields where you can modify the description, disposition protocol, and disposition of the port as shown in the following image. You cannot change the port name.


Edit Port

Choose parameters of the port that you wish to edit.

Port Name:	GETLIST
Description:	Created on 01/07/04.
Disposition Protocol:	FILE
Disposition:	file://c:\temp\x.txt

3. Make the required changes to the event port configuration fields and click *OK*.

Procedure: How to Delete an Event Port

To delete an event port:

1. Select the event port you want to delete.
2. In the right pane, move the pointer over *Operations* and select *Delete*.

   A confirmation dialog box opens, asking whether to delete the event port.
3. To delete the event port you selected, click *OK*.

The event port disappears from the list in the left pane.

Creating, Editing, or Deleting a Channel

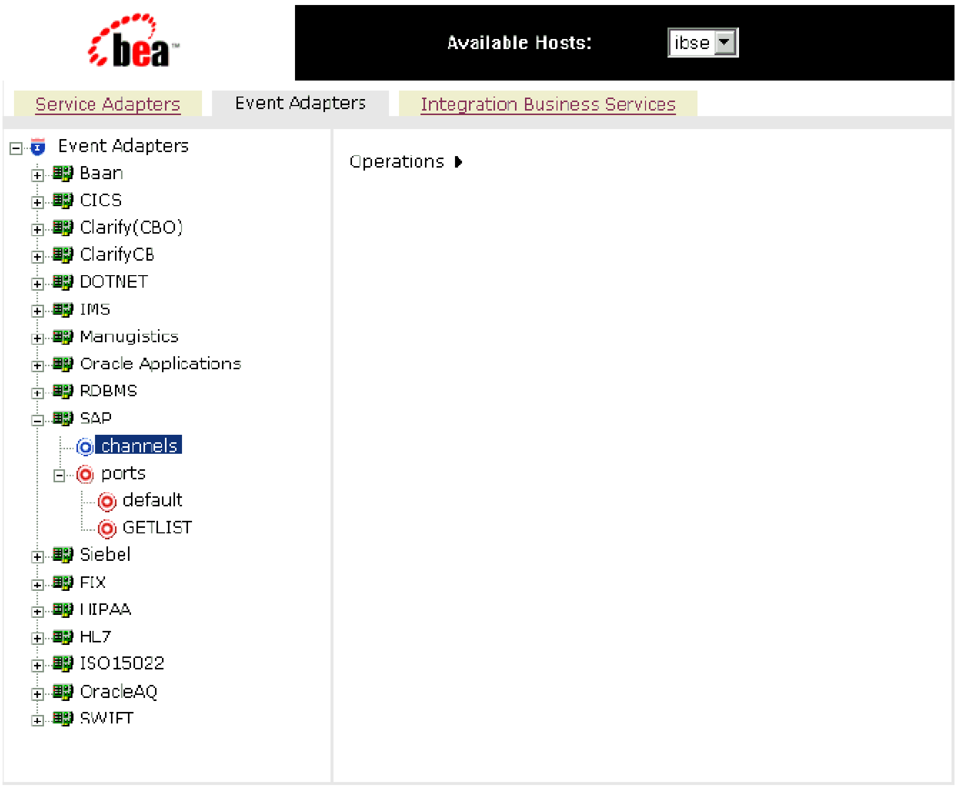
The following procedures describe how to create, edit, or delete a channel for your event adapter as well as how to start or stop a channel. All defined event ports must be associated with a channel. You can create a channel using Servlet Application Explorer.

Procedure: How to Create a Channel

To create a channel using Application Explorer:

1. Click the *Event Adapters* tab.

The list of adapters that support events appears in the left pane as shown in the following image.

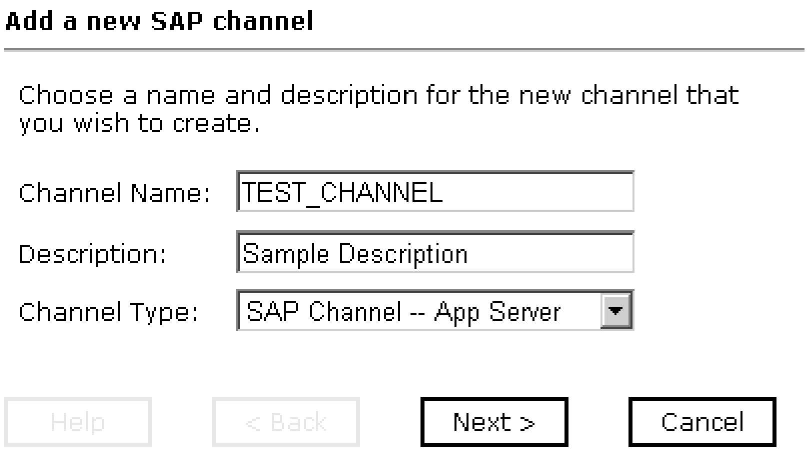


2. Expand the *Event Adapters* node, for example, SAP.

   The ports and channels nodes appear in the left pane.

3. Click the *channels* node.
4. In the right pane, move the pointer over *Operations* and select *Add a new channel*.

The Add a new SAP channel pane opens on the right as shown in the following image.



Choose a name and description for the new channel that you wish to create.

Channel Name: TEST_CHANNEL

Description: Sample Description

Channel Type: SAP Channel -- App Server

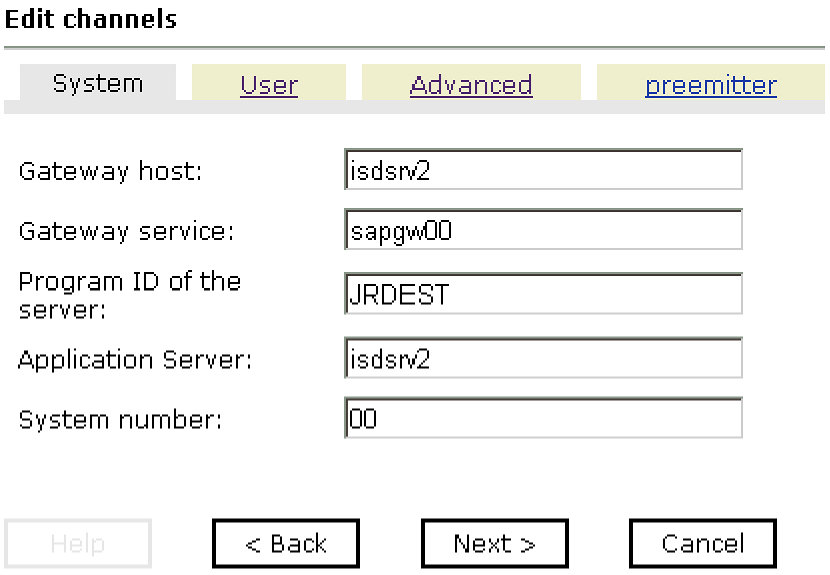
a. In the Channel Name field, type a name, for example, TEST_CHANNEL.

b. In the Description field, type a brief description (optional).

c. From the Channel Type drop-down list, select *SAP Channel -- App Server*.

5. Click *Next*.

The following graphic shows the Edit channels pane that opens on the right, with fields to enter a gateway host, a gateway service, the program ID of the server, an application server, and a system number for the channel.



Gateway host: isdsrv2

Gateway service: sapgw00

Program ID of the server: JRDEST

Application Server: isdsrv2

System number: 00

a. On the System tab, enter the information that is specific to your SAP system.

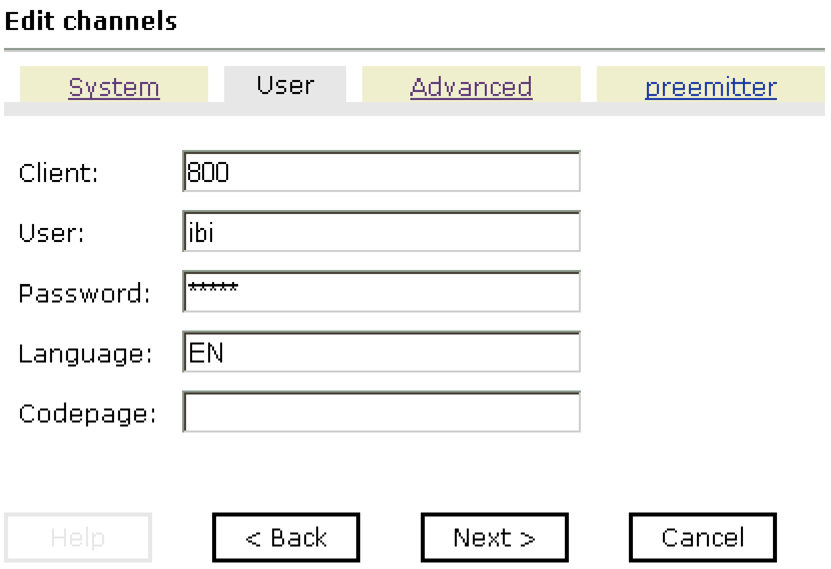
Note: The program ID of the server is case sensitive.

The following table lists and describes the system parameters.

Parameter	Description
Gateway host	Host name of the SAP gateway.
Gateway service	Service of the SAP gateway.
Program ID of the server	Program ID of the registered server program. **Note:** The program ID of the server is case sensitive.
Application Server	Host name of the SAP system.
System number	Number of the SAP system (two-digit numeric value).

b. Click the *User* tab.

The following graphic shows the User tab that opens, with fields to enter a client, a user, a password, a language, and a code page.



Client:	800
User:	ibi
Password:	*****
Language:	EN
Codepage:	

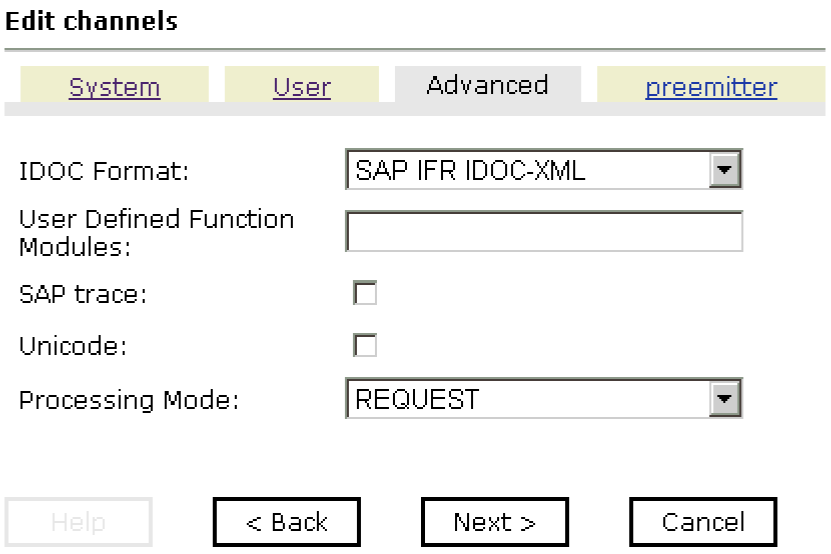
The following table lists and describes the user parameters.

Parameter	Description
Client	Client number defined for the SAP system for client communications.
User	Valid user ID for the SAP system.
Password	Valid password for the SAP system.
Language	Valid language key. EN (English) is the default value.
Codepage	Character code page value used by the SAP system (optional).

Note: For more information on obtaining the values found in the User tab, see your SAP system administrator.

c. Click the *Advanced* tab.

The following graphic shows the Advanced tab that opens, with fields to enter an IDoc format, any user defined function modules, check boxes to enable SAP traces or Unicode encoding of the event data, and a pull down menu with options for synchronous event processing.



IDOC Format: SAP IFR IDOC-XML

User Defined Function Modules:

SAP trace: ☐

Unicode: ☐

Processing Mode: REQUEST

d. Specify additional information or criteria for the channel you are creating.

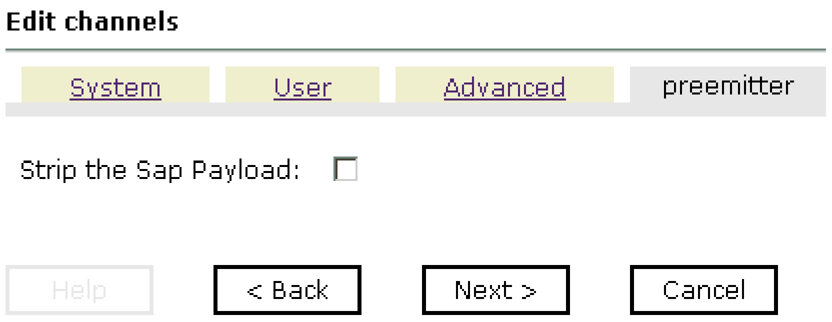
The following table lists and describes the parameters on the Advanced tab.

Parameter	**Description**
IDOC Format	The IDoc format to use for the channel you are creating. Possible values include: SAP IFR IDOC-XML (Default) FLAT IDOC
User Defined Function Modules	Path on the file system that points to your user-defined function module. A user-defined function module is used as a data source.
SAP trace	Select this option to enable SAP traces.
Unicode	Select this option if you want your response to have a Unicode format.
Processing Mode	Processing mode to use for your channel. Possible values include: REQUEST (Default) REQUEST_RESPONSE

Note: For more information on obtaining the values found in the Advanced tab, see your SAP system administrator.

6. Click the *preemitter* tab.

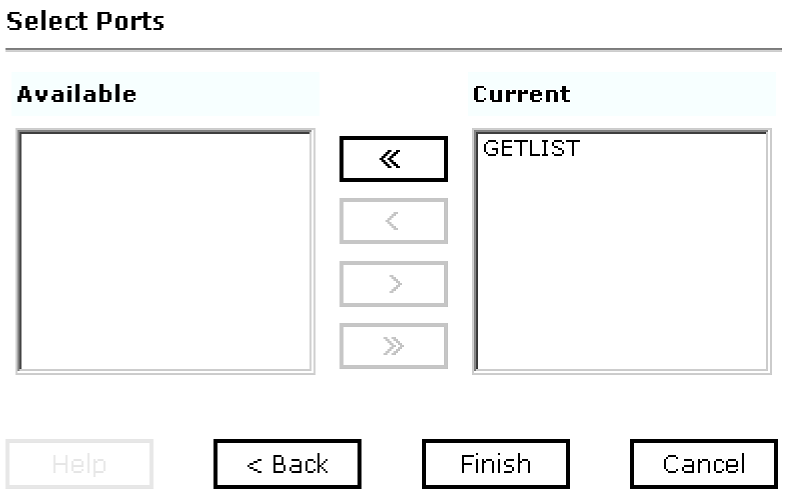
   The following graphic shows the preemitter tab that opens, with a checkbox that enables you to strip the SAP payload of an event document.



Strip the Sap Payload: ☐

7. Click *Next*.

The Select Ports pane opens as shown in the following image where you can move ports between the Current list and the Available list.

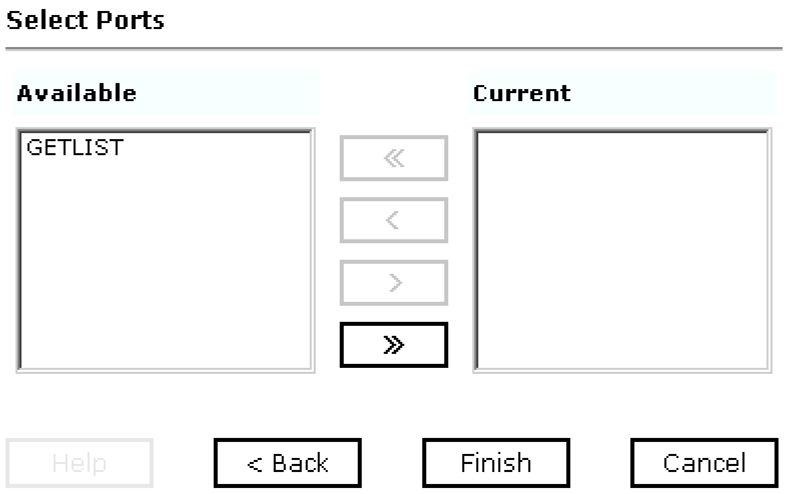


a. Select an event port from the list of current ports.

b. To transfer the port to the list of available ports, click the single left arrow button or to associate all event ports, click the double left arrow button.

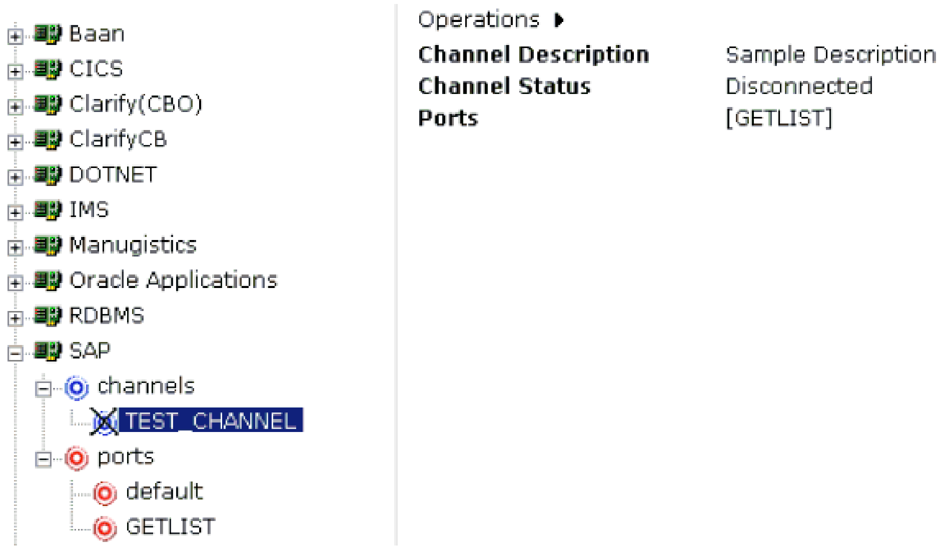
When only one port appears, it is as if you are transfering all ports. Therefore, only the double arrow button is active.

The port appears in the list of available ports as shown in the following image.



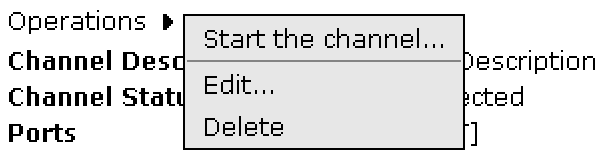
8. Click *Finish*.

   In the left pane, the TEST_CHANNEL is selected and has an 'X' over the icon. The summary pane opens on the right with the channel description, channel status, and available ports as shown in the following image.



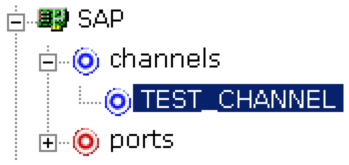
Channel Description	Sample Description
Channel Status	Disconnected
Ports	[GETLIST]

   All the information is associated with the channel you created. The 'X' over the icon indicates that the channel is currently disconnected. To activate your event configuration, you must start the channel using the Operations menu, as shown in the following image.



9. Move the pointer over *Operations* and select *Start the channel*.

The channel you created is now active, and the 'X' disappears, as shown in the following image.



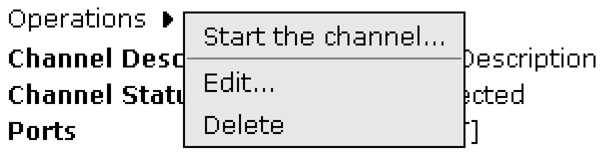
10. To stop the channel at any time, move the pointer over *Operations* and select *Stop the channel*.

Procedure: How to Edit a Channel

To edit an existing channel:

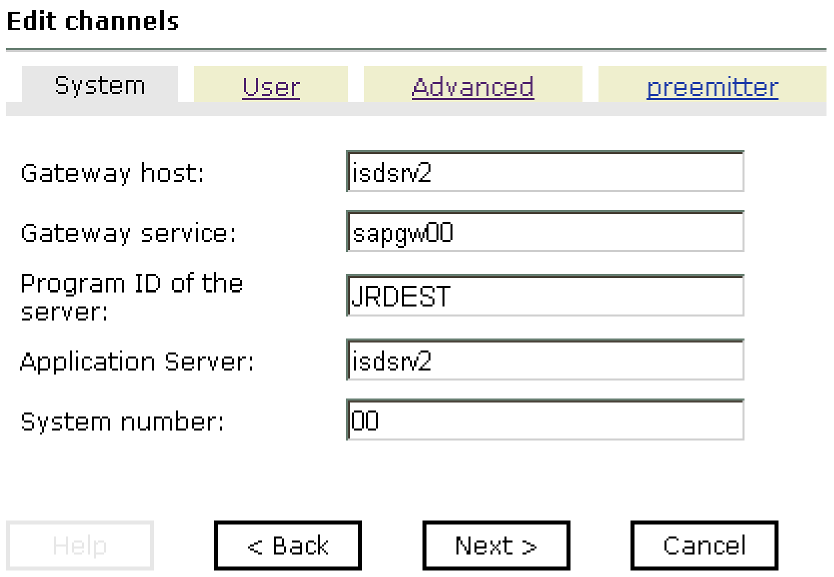
1. In the left pane, select the channel you want to edit.

   The Operations menu opens in the right pane over the summary of the channel information, as shown in the following image.



2. Move the pointer over *Operations* and select *Edit*.

The Edit channels pane opens on the right as shown in the following image, with the User tab active.



Edit channels

Gateway host:	isdsrv2
Gateway service:	sapgw00
Program ID of the server:	JRDEST
Application Server:	isdsrv2
System number:	00

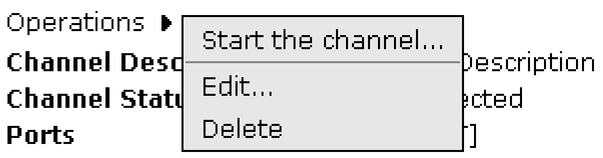
3. Make the required changes to the channel configuration fields.
4. To continue editing information on the other tabs, click *Next*.
5. When you complete your edits, click *Finish*.

Procedure: How to Delete a Channel

To delete an existing channel:

1. In the left pane, select the channel you want to delete.

   The Operations menu opens in the right pane over the summary of the channel information, as shown in the following image.



2. Move the pointer over *Operations* and select *Delete*.

   A confirmation dialog box opens asking if you want to delete this item.

3. To delete the channel you selected, click *OK*.

   The channel disappears from the list in the left pane.

Synchronous Event Processing

You can configure synchronous event processing using Application Explorer to trigger a Web service after an event occurs in the SAP system. The event response that is received can then be routed to another disposition for further processing.

Procedure: How to Configure Synchronous Event Processing Using Application Explorer

To configure synchronous event processing:

1. Create a Web service for an SAP Remote Function Module, for example, BAPI_MATERIAL_GETLIST.
2. View the WSDL file.
3. In the Create New Port pane, create a port using the SOAP disposition.
   a. In the Name field, type a name.
   b. In the Description field, type a brief description (optional).
   c. From the Disposition Protocol drop-down list, select *SOAP*.
   d. In the Disposition field, enter a SOAP destination, using the following format:

```
soap:[wsdl-url];soapaction=[myaction];
method=[web service method];namespace=[namespace];
responseTo=[pre-defined port name or another disposition URL];
errorTo=[pre-defined port name or another disposition url]
```

The following table lists and describes the parameters for the disposition.

Parameter	**Description**
wsdl-url	The URL to the WSDL file that is required to create the SOAP message, for example: `http://localhost:7001/ibse/IBSEServlet/test/`*`webservice`*`.ibs?wsdl` where: *`webservice`* Is the name of the Web service you created using Application Explorer. To find this value, you can navigate to the Integration Business Services tab and open the Service Description link in a new window. The WSDL URL appears in the Address field. Alternatively, you can open the WSDL file in a third-party XML editor (for example, XMLSPY) and view the SOAP request settings.
soapaction	Method that is called by the SOAP disposition. This value can be found in the WSDL file.
method	Web service method you are using. This value is in the WSDL file.
namespace	The XML namespace you are using. This value is in the WSDL file.
responseTo	Location to which responses are posted. Can be a predefined port name or another URL. Optional. The URL must be complete, including the protocol.
errorTo	Location where error documents are sent. This can be a predefined port name or another full URL. Optional.

Note: To use the SOAP disposition with a synchronous event, use Remote Function Modules to generate the schema and WSDL file instead of using the Business Object Repository for any RFC or BAPI.

The following is an example of a completed SOAP disposition:

```
soap:http://localhost:7001/ibse/IBSEServlet/test/soapWS.ibs?wsdl;
soapaction=soapWS.GETLISTRequest@test@@;method=GETLIST;
namespace=urn:iwaysoftware:ibse:jul2005:GETLIST;
responseTo=ifile://c:\output\sap\soapOut.xml
```

4. Using the Edit channels pane, create a channel.
   a. Provide the required information to connect to SAP in the System and User tabs.
   b. In the Advanced tab, from the Processing Mode drop-down list, select *REQUEST_RESPONSE*.
5. Associate the port you created earlier with the new channel.
6. Start the channel.

   A Web service for an SAP Remote Function Module, for example, BAPI_MATERIAL_GETLIST, is triggered after an event occurs in the SAP system. The response document is returned and routed to a file location.

CHAPTER 6

Using Integration Business Services Policy-Based Security

Topics:

- Integration Business Services Policy-Based Security
- Configuring Integration Business Services Policy-Based Security

The following topics describe how Integration Business Services policy-based security works and how to configure it using the Servlet Application Explorer.

Before you can configure policy-based security, you must enable security for your environment through the Integration Business Services Engine (iBSE) configuration page.

For more information, see the *BEA WebLogic ERP Adapter Installation and Configuration* manual.

Note: BEA recommends that you leave policy-based security disabled.

Integration Business Services Policy-Based Security

Integration Business Services provide a layer of abstraction between the back-end business logic they invoke and the user or application running the business service. This enables easy application integration but raises the issue of controlling the use and execution of critical and sensitive business logic that is run as a business service.

Servlet Application Explorer controls the use of business services that use adapters with a feature called policy-based security. This feature enables an administrator to apply policies to Integration Business Services (iBS) to deny or permit their execution.

A *policy* is a set of privileges associated with the execution of a business service that can be applied to an existing or new iBS. When you assign specific rights or privileges inside a policy, you need not recreate privileges for every iBS that has security issues in common with other Integration Business Services. Instead, you can use one policy for many Integration Business Services.

The goal is to secure requests at both the transport and the SOAP request level that are transmitted on the wire. Some policies do not deal with security issues directly but affect the run-time behavior of the business services to which they are applied.

The Integration Business Services Engine (iBSE) administrator creates an instance of a policy type, names it, associates individual users and/or groups (a collection of users), and then applies the policy to one or more business services.

You can assign a policy to an iBS or to a method within an iBS. If a policy is applied only to a method, other methods in that iBS are not governed by it. However, if a policy is applied to the iBS, all methods are governed by it. At run time, the user ID and password that are sent to iBSE in the SOAP request message are checked against the list of users for all policies applied to the specific iBS. The Resource Execution policy type is supported and dictates who can or cannot execute the iBS.

When a policy is not applied, the default value for an iBS is to "grant all." For example, anyone can execute the iBS until the Resource Execution policy is associated to the iBS. At that time, only users granted execution permission, or those who do not belong to a group that was denied execution permissions, have access to the iBS.

Configuring Integration Business Services Policy-Based Security

Before you create instances of policies, you must have a minimum of one user or one group to associate to an instance. You can create users and groups using Servlet Application Explorer. For more information, see *How to Create a User to Associate With a Policy* on page 6-3 or *How to Create a Group to Associate With a Policy* on page 6-5.

An execution policy governs who can execute the business service to which the policy is applied. For more information, see *How to Create an Execution Policy* on page 6-7.

You configure the IP and Domain Restriction policy type slightly differently from other policy types. The IP and Domain Restriction policy type controls connection access to Integration Business Services Engine (iBSE) and therefore, need not be applied to an individual business service. You need not create a policy, however, you must enable the Security Policy option in Servlet Application Explorer. For more information, see *How to Configure IP and Domain Restrictions* on page 6-10.

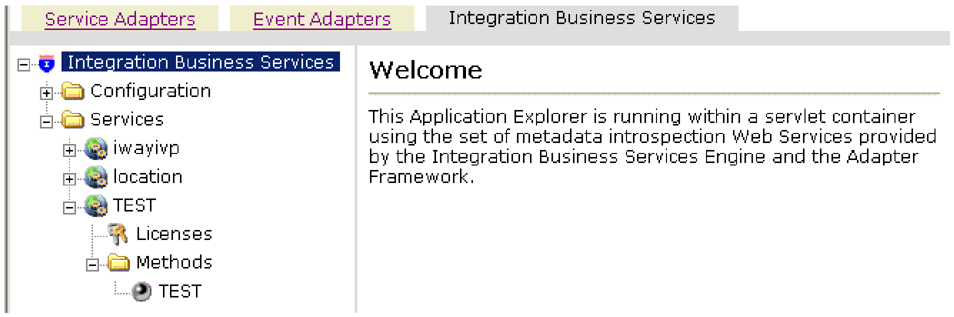
Note: BEA recommends that you leave policy-based security disabled.

Procedure: How to Create a User to Associate With a Policy

To create a user to associate with a policy:

1. Open Servlet Application Explorer.

   The following image shows the window that opens and includes three tabs corresponding to Service Adapters, Event Adapters, and Integration Business Services. The Integration Business Services tab is active and displays a Welcome screen on the right and the Integration Business Services node expanded on the left.

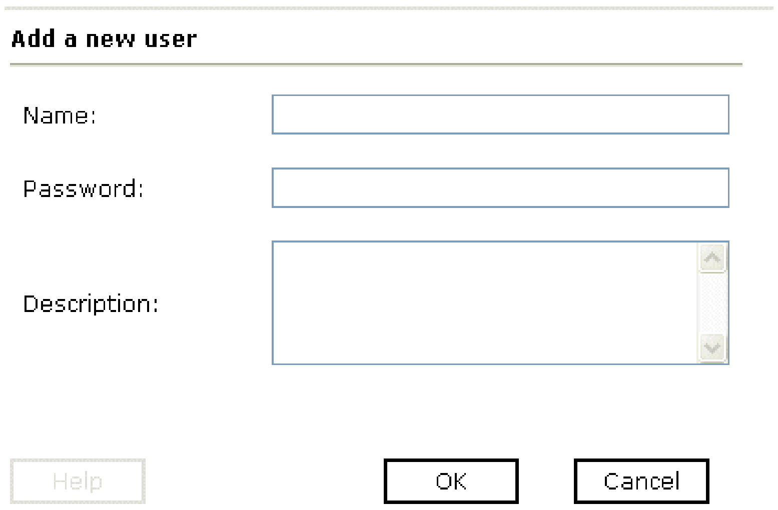


This Application Explorer is running within a servlet container using the set of metadata introspection Web Services provided by the Integration Business Services Engine and the Adapter Framework.

   a. Click the *Integration Business Services* tab.
   b. Expand the *Configuration* node.
   c. Expand the *Security* node.
   d. Expand the *Users and Groups* node.
   e. Select *Users*.

2. In the right pane, move the pointer over *Operations* and select *Add*.

The Add a new user pane opens as shown in the following image.



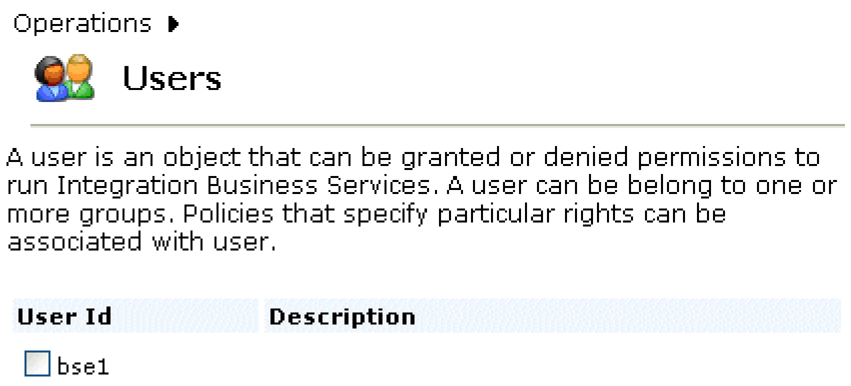
a. In the Name field, type a user ID.

b. In the Password field, type the password associated with the user ID.

c. In the Description field, type a description of the user (optional).

3. Click *OK.*

   The Users pane opens, which includes the definition of a user and a user ID and description, identifying that a new user was added to the configuration, as shown in the following image.



Users

A user is an object that can be granted or denied permissions to run Integration Business Services. A user can be belong to one or more groups. Policies that specify particular rights can be associated with user.

User Id	Description
☐ bse1	

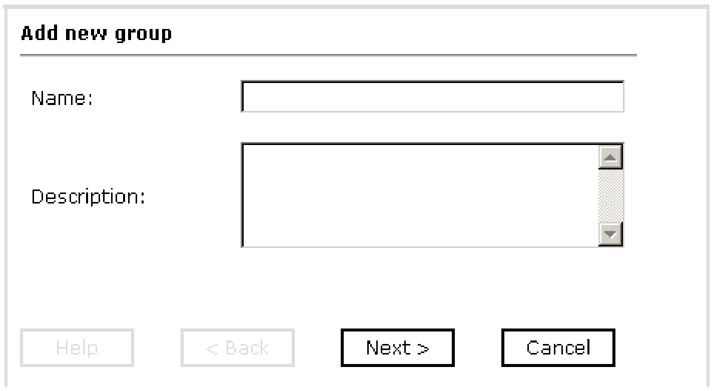
Procedure: How to Create a Group to Associate With a Policy

To create a group to associate with a policy:

1. Open Servlet Application Explorer.
   - a. Click the *Integration Business Services* tab.
   - b. Expand the *Configuration* node.
   - c. Expand the *Security* node.
   - d. Expand the *Users and Groups* node.
   - e. Select *Groups.*

2. In the right pane, move the pointer over *Operations* and click *Add*.

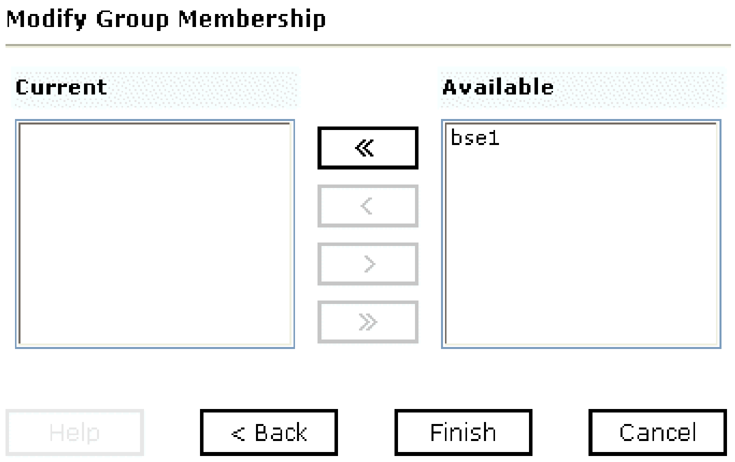
   The Add new group pane opens, as shown in the following image.



   a. In the Name field, type a a name for the group.
   b. In the Description field, type a description for the group (optional).

3. Click *Next*.

   The Modify Group Membership pane opens where you can move users between the Current and Available lists, as shown in the following image.

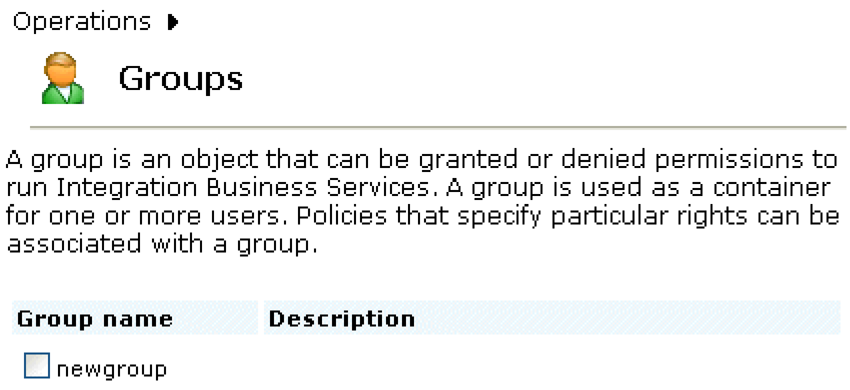


4. Either highlight a single user in the list of available users and add it to the current list by clicking the left arrow, or click the double left arrow to add all users in the list of available users to the group.

5. After you select a minimum of one user, click *Finish*.

   The new group is added.

   The Groups pane opens, which includes a definition of a group and the group name and description, identifying that a new group was added to the configuration, as shown in the following image.



A group is an object that can be granted or denied permissions to run Integration Business Services. A group is used as a container for one or more users. Policies that specify particular rights can be associated with a group.

Group name	Description
newgroup	

Procedure: How to Create an Execution Policy

To create an execution policy:

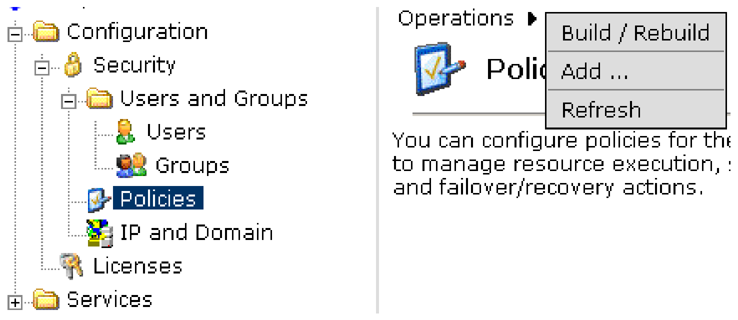
1. Open Servlet Application Explorer.

   a. Click the *Integration Business Services* tab.

   b. Expand the *Configuration* node.

   c. Select *Policies*.

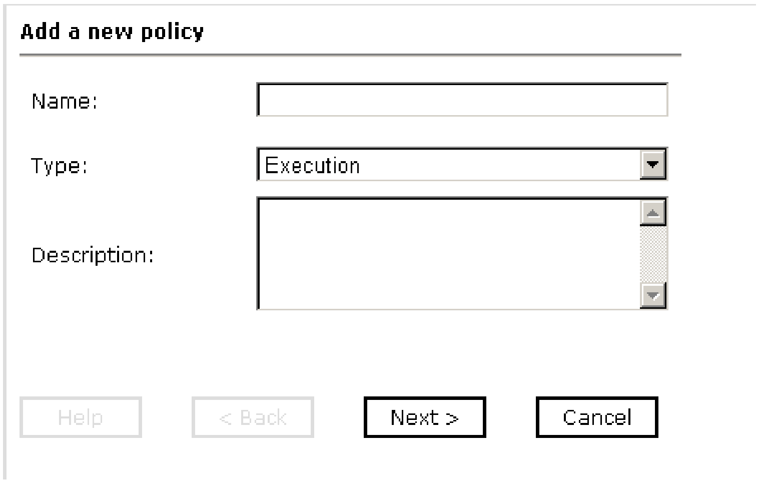
   The Policies pane opens on the right where you can apply a policy as shown in the following image. The Operations menu displays three options, Build/Rebuild, Add, and Refresh.



You can configure policies for the
to manage resource execution,
and failover/recovery actions.

2. Move the pointer over *Operations* and select *Add*.

   The Add a new policy pane opens as shown in the following image.



Add a new policy

Name:

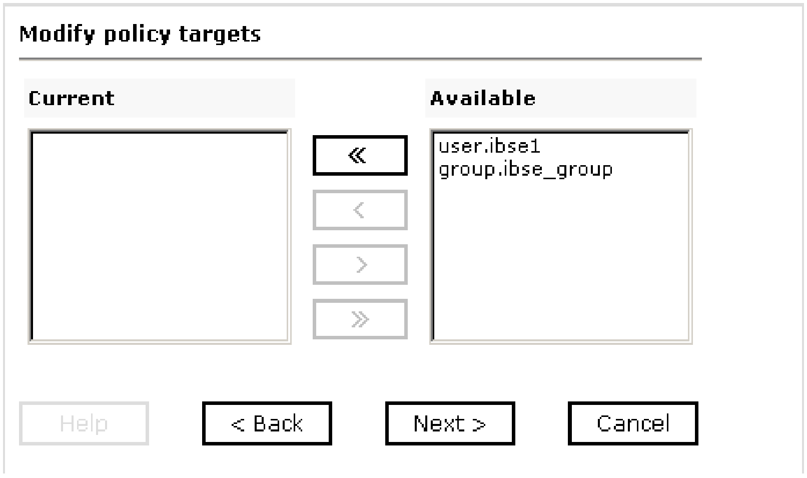
Type: Execution

Description:

   a. In the Name field, type a name for the policy.
   b. From the Type drop-down list, select *Execution*.
   c. In the Description field, type a description for the policy (optional).

3. Click *Next*.

   The Modify policy targets pane opens where you can move targets between the Current and Available lists, as shown in the following image.



Modify policy targets

Current

Available

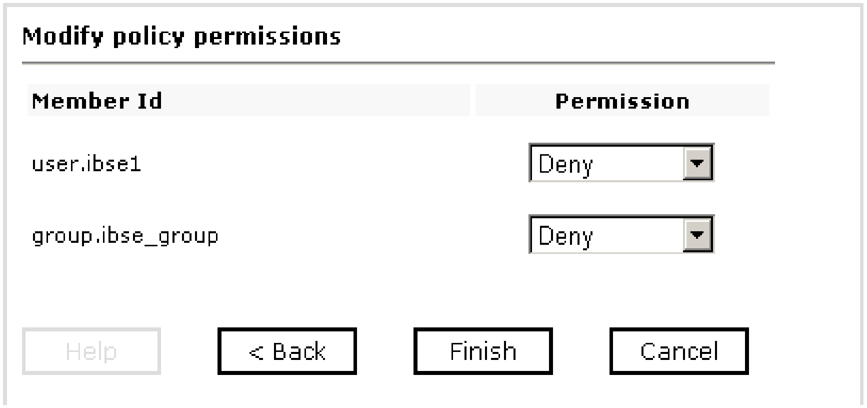
user.ibse1
group.ibse_group

4. Select a minimum of one user or group from the Available pane.

   Note: This user ID is checked against the value in the user ID element of the SOAP header sent to iBSE in a SOAP request.

5. Click *Next*.

   The following image shows the Modify policy permissions pane that opens and includes drop-down lists where you can select to grant or deny permission to members.



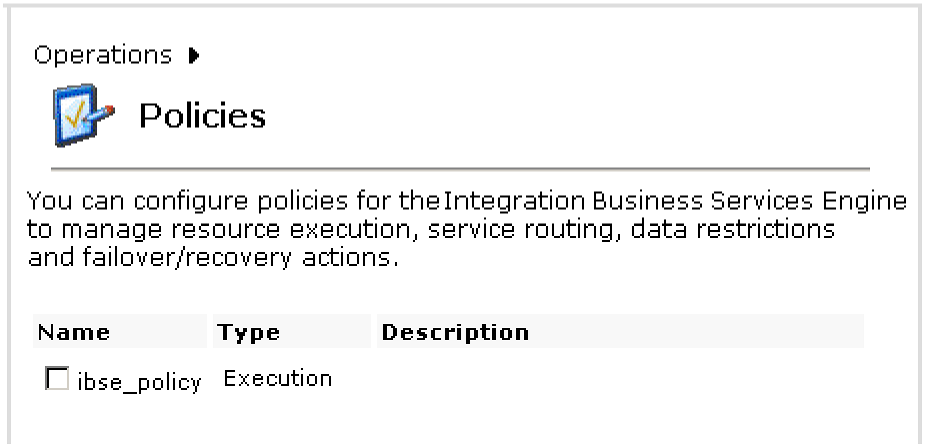
Modify policy permissions

Member Id	Permission
user.ibse1	Deny
group.ibse_group	Deny

6. To assign whether users or groups may execute the Integration Business Services, select *Grant* to permit execution or *Deny* to restrict execution from a Permission drop-down list.

7. Click *Finish*.

A Policies pane opens that summarizes your configuration and includes a definition of policies and the name, type, and description of the policies, as shown in the following image.



You can configure policies for the Integration Business Services Engine to manage resource execution, service routing, data restrictions and failover/recovery actions.

Name	Type	Description
☐ ibse_policy	Execution	

Procedure: How to Configure IP and Domain Restrictions

To configure IP and domain restrictions:

1. Open Servlet Application Explorer.

a. Select the *Integration Business Services* tab.

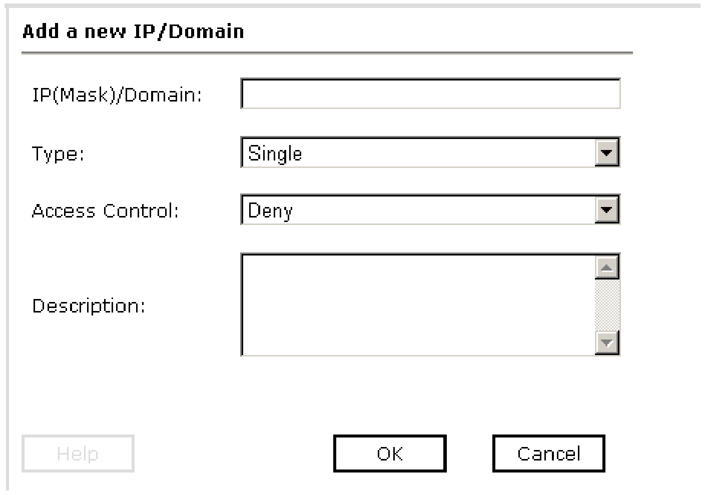
b. Expand the *Configuration* node.

c. Expand the *Security* node.

d. Select *IP and Domain*.

2. In the right pane, move the pointer over *Operations* and click *Add*.

   The Add a new IP/Domain pane opens where you enter information for the IP/Domain in four fields, as shown in the following image. You must select a type of restriction from a drop-down list before you can enter information in the IP(Mask)/Domain field.



Add a new IP/Domain

IP(Mask)/Domain:

Type: Single

Access Control: Deny

Description:

   a. From the Type drop-down list, select the type of restriction.

   b. In the IP(Mask)/Domain field, type the IP or domain name using the following guidelines.

      If you select **Single** (Computer) from the Type drop-down list, you must provide the IP address for that computer. If you only know the DNS name for the computer, click *DNS Lookup* to obtain the IP Address based on the DNS name.

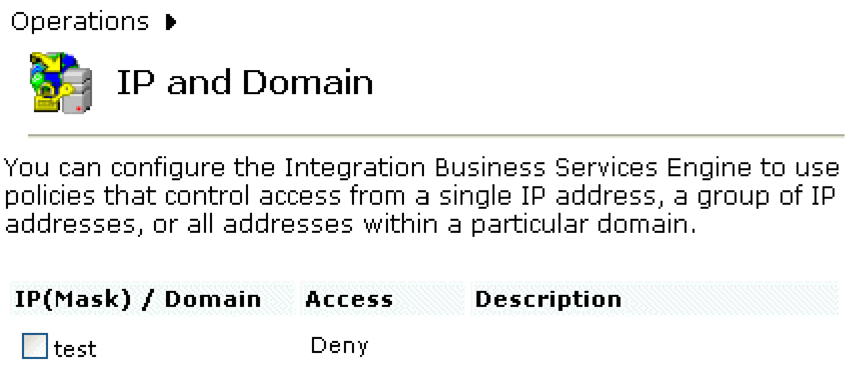
      If you select **Group** (of Computers), you must provide the IP address and subnet mask for the computer group.

      If you select **Domain**, you must provide the domain name, for example, yahoo.com.

3. From the Access Control drop-down list, select *Grant* to permit access or *Deny* to restrict access for the IP addresses and domain names you are adding.

4. Click *OK*.

The IP and Domain pane opens and summarizes your configuration including the domain name, whether access is granted or denied, and a description (optional), as shown in the following image.



You can configure the Integration Business Services Engine to use policies that control access from a single IP address, a group of IP addresses, or all addresses within a particular domain.

IP(Mask) / Domain	Access	Description
☐ test	Deny	

CHAPTER 7

Management and Monitoring

Topics:

After you create services and events using Application Explorer, you can use management and monitoring tools to measure the performance in your run-time environment. This section describes how to configure and use these features.

Managing and Monitoring Services and Events Using iBSE

iWay Business Services Engine (iBSE) provides a console that enables you to manage and monitor services and events currently in use. The console also displays resource usage and invocation statistics. These indicators can help you adjust your environment for optimum efficiency.

The following monitoring levels are available for services:

- System
- Service
- Method

The following monitoring levels are available for events:

- System
- Channel
- Port

Procedure: How to Configure Monitoring Settings

To configure monitoring settings:

1. Ensure that your Application Server is started.
2. To access the monitoring console, enter the following URL in your Web browser:

   `http://`*`hostname`*`:`*`port`*`/ibse/IBSEConfig`

   where:

   `hostname`

   Is the machine where the application server is running.

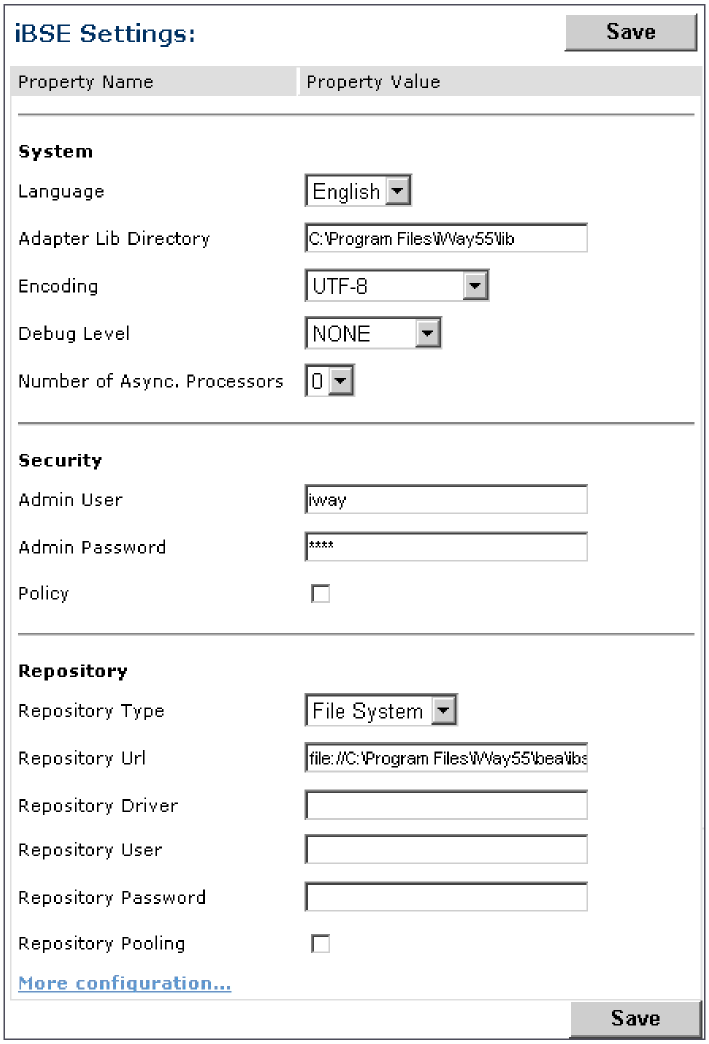
   `port`

   Is the HTTP port for the application server.

   The iBSE Settings window opens as shown in the following image. It consists of three panes. To configure system settings, the System pane includes drop-down lists for selecting language, encoding, the debug level, and the number of asynchronous processors. It also contains a field where you can enter a path to the adapter lib directory.

   To configure security settings, the Security pane contains fields for typing the Admin User name and the associated password and a check box for specifying policy.

To configure repository settings, the Repository pane contains a drop-down list for selecting the repository type; fields to type information for the repository URL, driver, user, and password; and a check box where you can enable repository pooling. In the upper and lower right of the window is a Save button. In the lower left of the window is an option to access more configuration settings.



Property Name	Property Value
System	
Language	English
Adapter Lib Directory	C:\Program Files\iWay55\lib
Encoding	UTF-8
Debug Level	NONE
Number of Async. Processors	0
Security	
Admin User	iway
Admin Password	****
Policy	☐
Repository	
Repository Type	File System
Repository Url	file://C:\Program Files\iWay55\bea\ibs
Repository Driver	
Repository User	
Repository Password	
Repository Pooling	☐

3. Click *More configuration*.

   Tip: To access the monitoring console directly, enter the following URL in your Web browser:

   `http://hostname:port/ibse/IBSEStatus`

   where:

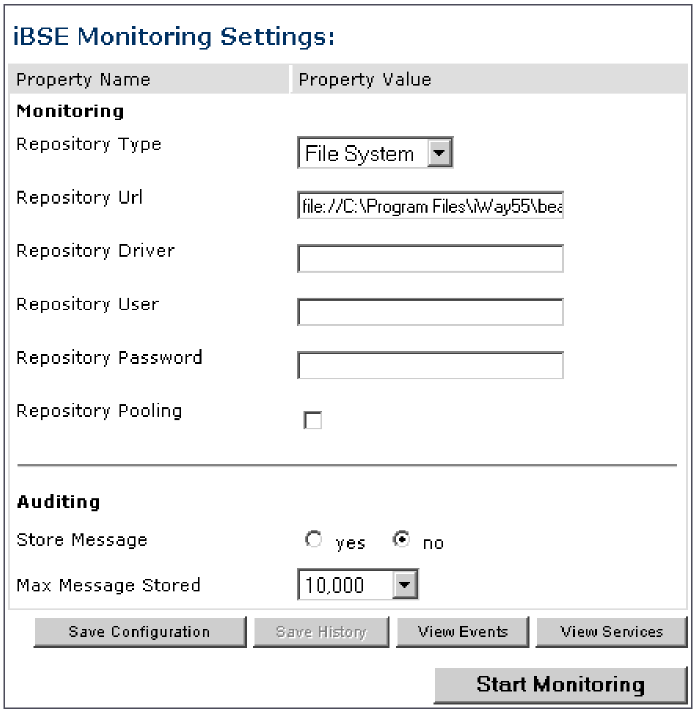
   `hostname`

   Is the machine where the application server is running.

   `port`

   Is the HTTP port for the application server.

   The iBSE Monitoring Settings window which is divided into two panes opens as shown in the following image. At the bottom of the window is a row of command buttons that enable you to save your configuration, view events, or view services. The Save History button is inactive.



iBSE Monitoring Settings:

Property Name	Property Value
Monitoring	
Repository Type	File System
Repository Url	file://C:\Program Files\iWay55\bea
Repository Driver	
Repository User	
Repository Password	
Repository Pooling	☐
Auditing	
Store Message	yes [✓ no]
Max Message Stored	10,000

   a. In the Monitoring pane, from the Repository Type drop-down list, select the type of repository you are using.

   b. To connect to the database in the Repository Url field, type a JDBC URL.

   - **c.** To connect to the database in the Repository Driver field, type a JDBC Class.
   - **d.** To access the monitoring repository database, type a user ID and password.
   - **e.** To enable pooling, select the *Repository Pooling* check box.
   - **f.** In the Auditing pane, click *yes* if you want to store messages.

     This option is disabled by default.

     Note: You must start and then stop monitoring to enable this option.
   - **g.** From the drop-down list, select the maximum number of messages to store.

     By default, 10,000 is selected.

     Note: Depending on your environment and the number of messages that are exchanged, storing a large number of messages may affect system performance. If you require more information about your system resources, consult your system administrator.
   - **h.** Click *Save Configuration*.

4. Click *Start Monitoring*.

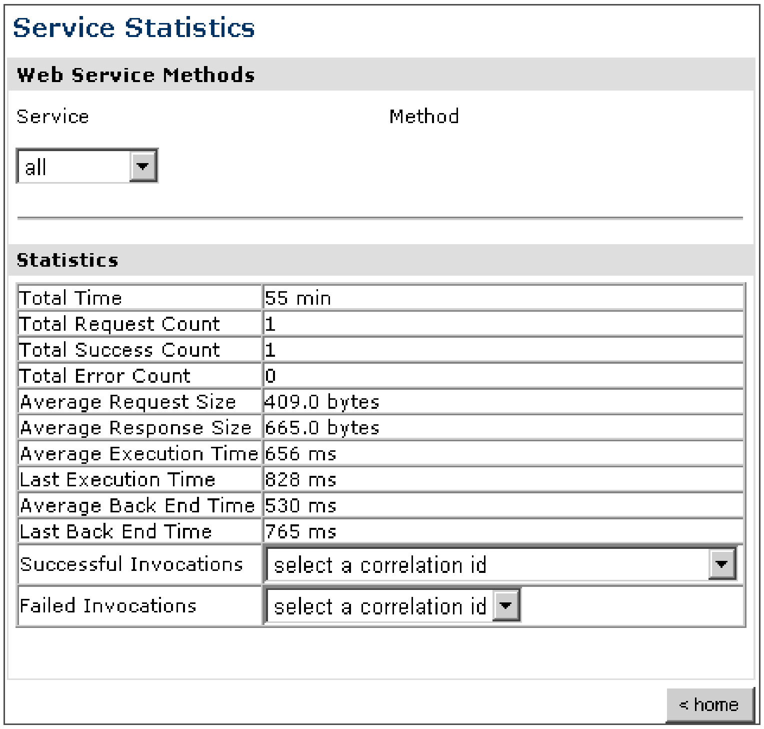
   iBSE begins to monitor all services and events currently in use. If you selected the option to store messages, iBSE stores messages.
5. To stop monitoring, click *Stop Monitoring*.

Procedure: How to Monitor Services

To monitor services:

1. Ensure that your Application Server is started.
2. From the iBSE Monitoring Settings window, click *Start Monitoring*.
3. Click *View Services*.

   The System Level Summary (Service Statistics) window opens, as shown in the following image. The Web Service Methods pane contains a drop-down list where you select a service. On the right, space is reserved for a drop-down list of methods that will appear. The Statistics pane contains a table with a summary of service statistics and two drop-down lists where you can select a successful or failed invocation to view more information about that service. At the bottom of the window is a home button to click to return to the iBSE Monitoring Settings window.



Total Time	55 min
Total Request Count	1
Total Success Count	1
Total Error Count	0
Average Request Size	409.0 bytes
Average Response Size	665.0 bytes
Average Execution Time	656 ms
Last Execution Time	828 ms
Average Back End Time	530 ms
Last Back End Time	765 ms
Successful Invocations	select a correlation id
Failed Invocations	select a correlation id

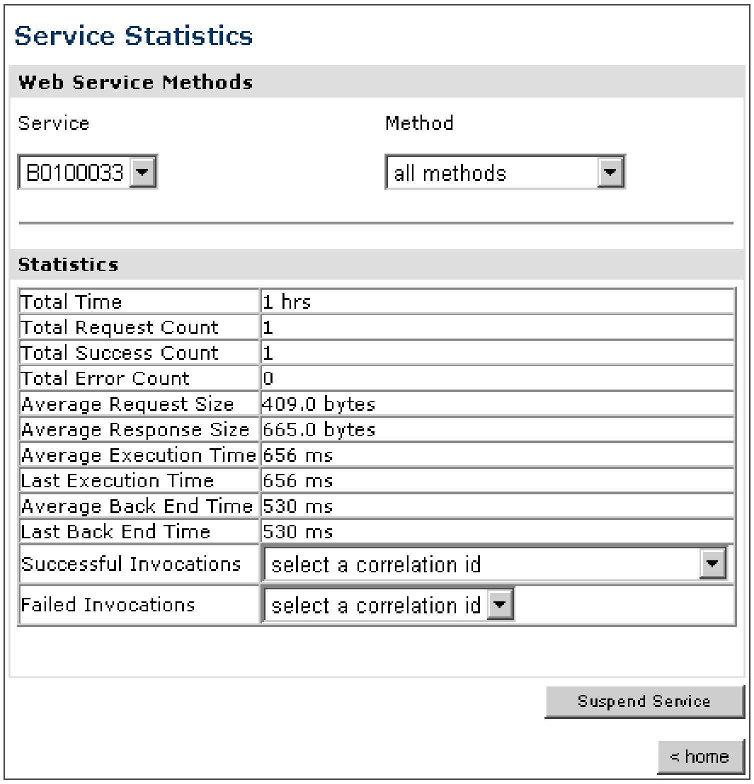
The system level summary provides services statistics at a system level.

The following table lists and describes each service statistic.

Statistic	**Description**
Total Time	Total amount of time iBSE monitors services. The time starts after you click Start Monitoring in the iBSE Monitoring Settings window.
Total Request Count	Total number of services requests made during the monitoring session.
Total Success Count	Total number of successful service executions.
Total Error Count	Total number of errors encountered.
Average Request Size	Average size of an available service request.
Average Response Size	Average size of an available service response.
Average Execution Time	Average execution time for a service.
Last Execution Time	Last execution time for a service.
Average Back End Time	Average back-end time for a service.
Last Back End Time	Last back-end time for a service.
Successful Invocations	Successful services arranged by correlation ID. To retrieve more information for a service, select the service from the drop-down list.
Failed Invocations	Failed services arranged by correlation ID. To retrieve more information for a service, select the service from the drop-down list.

4. Select a service from the drop-down list.

   The System Level Summary (Service Statistics) window opens as shown in the following image. The Web Service Methods pane contains a drop-down list on the left where you select a service and a drop-down list on the right where you select a service method. The Statistics pane contains a table with a summary of service statistics and two drop-down lists. To view more information about that service, you can select it from the Successful Invocations or Failed Invocations drop-down list. To suspend or resume a service, you can click a button in the lower right. To return to the iBSE Monitoring Settings window, you click the home button.



Service Statistics

Web Service Methods

Service: B0100033

Method: all methods

Statistics

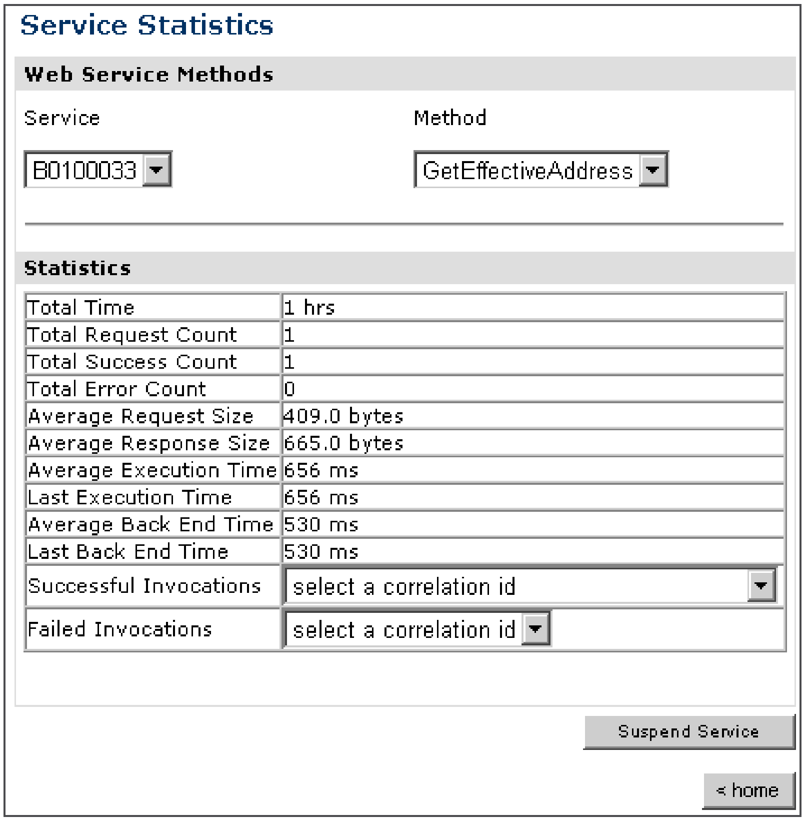
Total Time	1 hrs
Total Request Count	1
Total Success Count	1
Total Error Count	0
Average Request Size	409.0 bytes
Average Response Size	665.0 bytes
Average Execution Time	656 ms
Last Execution Time	656 ms
Average Back End Time	530 ms
Last Back End Time	530 ms
Successful Invocations	select a correlation id
Failed Invocations	select a correlation id

   a. To stop a service at any time, click *Suspend Service*.

   b. To restart the service, click *Resume Service*.

5. Select a method for the service from the Method drop-down list.

The Method Level Summary (Service Statistics) window opens as shown in the following image. The Web Service Methods pane contains a drop-down list on the left where you select a service and a drop-down list on the right where you select a service method. The Statistics pane contains a table with a summary of service statistics and two drop-down lists. To view more information about that service, you can select it from the Successful Invocations or Failed Invocations drop-down list. To suspend or resume a service, you can click a button in the lower right. To return to the iBSE Monitoring Settings window, you click the home button.



Service Statistics

Web Service Methods

Service: B0100033

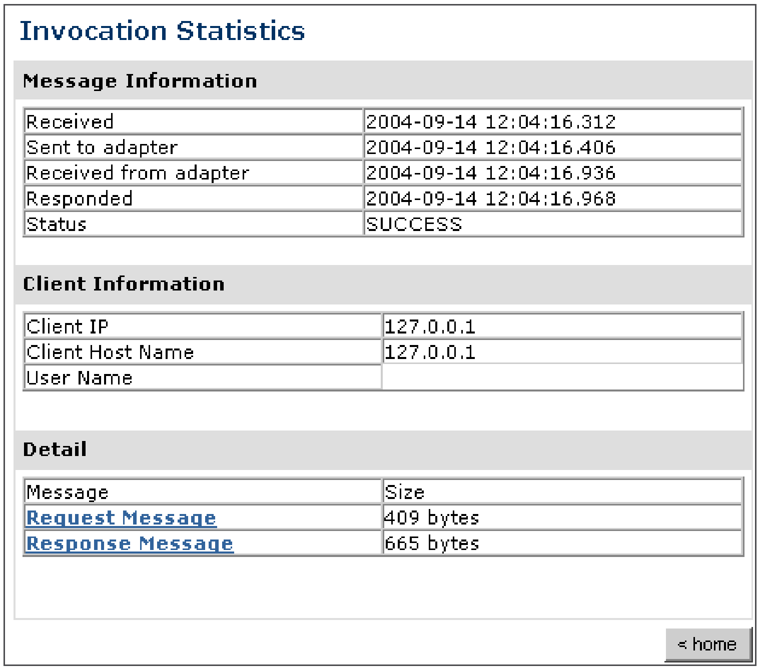
Method: GetEffectiveAddress

Statistics

Total Time	1 hrs
Total Request Count	1
Total Success Count	1
Total Error Count	0
Average Request Size	409.0 bytes
Average Response Size	665.0 bytes
Average Execution Time	656 ms
Last Execution Time	656 ms
Average Back End Time	530 ms
Last Back End Time	530 ms
Successful Invocations	select a correlation id
Failed Invocations	select a correlation id

6. For additional information about a successful service and its method, select a service based on its correlation ID from the Successful Invocation drop-down list.

   The Invocation Level Statistics window opens as shown in the following image. The Message Information pane contains a table of information about the message. The Client Information pane contains a table of information about the client. The Detail pane contains a table that shows the size of the request and response messages.



Invocation Statistics

Message Information

Received	2004-09-14 12:04:16.312
Sent to adapter	2004-09-14 12:04:16.406
Received from adapter	2004-09-14 12:04:16.936
Responded	2004-09-14 12:04:16.968
Status	SUCCESS

Client Information

Client IP	127.0.0.1
Client Host Name	127.0.0.1
User Name	

Detail

Message	Size
Request Message	409 bytes
Response Message	665 bytes

7. To view the XML request document in your Web browser, click *Request Message*.

   You can also view the XML response document for the service.

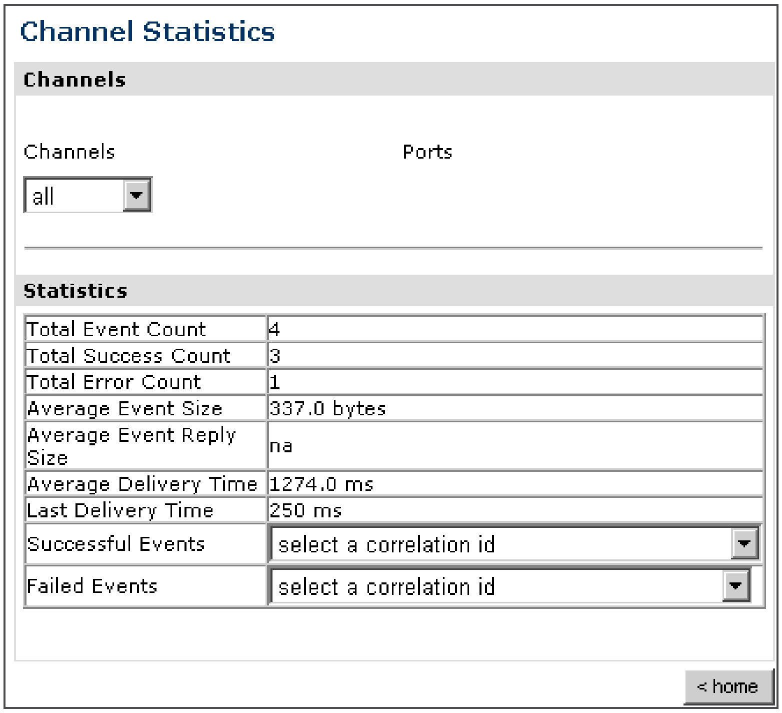
8. To return to the iBSE Monitoring Settings window, click *home*.

Procedure: How to Monitor Events

To monitor events:

1. Ensure that your Application Server is started.
2. In the iBSE Monitoring Settings window, click *Start Monitoring*.
3. Click *View Events*.

   The System Level Summary (Channel Statistics) window opens as shown in the following image. The Channels pane contains a drop-down list on the left where you select a channel. On the right, space is reserved for a drop-down list of ports that will appear. The Statistics pane contains a table with a summary of event statistics and two drop-down lists where you can select a successful or failed event to view more information about that event. In the lower right of the window is a home button.



Channel Statistics

Channels

Channels: all

Ports

Statistics

Total Event Count	4
Total Success Count	3
Total Error Count	1
Average Event Size	337.0 bytes
Average Event Reply Size	na
Average Delivery Time	1274.0 ms
Last Delivery Time	250 ms
Successful Events	select a correlation id
Failed Events	select a correlation id

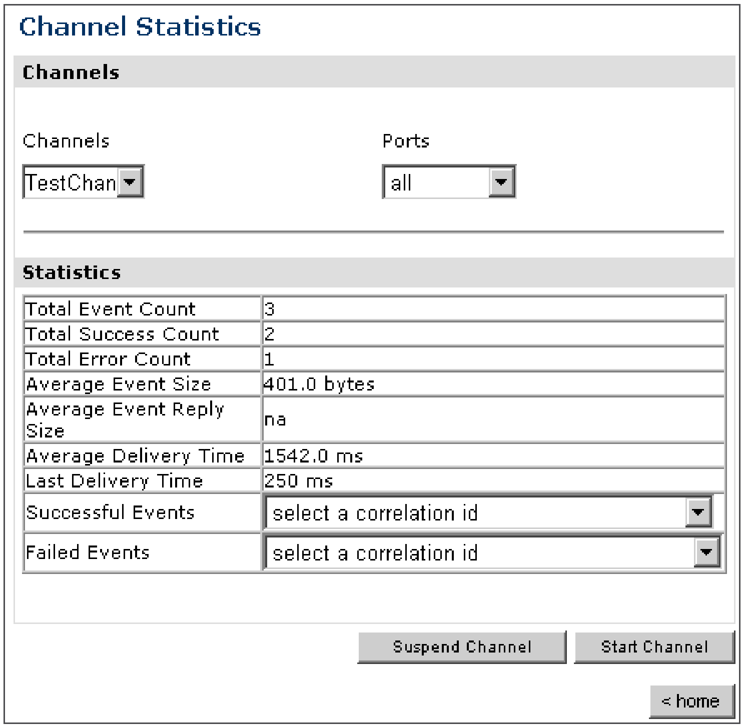
The system level summary provides event statistics at a system level.

The following table lists and describes each event statistic.

Statistic	**Description**
Total Event Count	Total number of events.
Total Success Count	Total number of successful event executions.
Total Error Count	Total number of errors encountered.
Average Event Size	Average size of an available event request.
Average Event Reply Size	Average size of an available event response.
Average Delivery Time	Average delivery time for an event.
Last Delivery Time	Last delivery time for an event.
Successful Events	Successful events arranged by correlation ID. To retrieve more information for an event, select the event from the drop-down list.
Failed Events	Failed events arranged by correlation ID. To retrieve more information for an event, select the event from the drop-down list.

4. Select a channel from the drop-down list.

The Channel Level Event Summary (Channel Statistics) window opens as shown in the following image. The Channels pane contains a drop-down list on the left where you select a channel and a drop-down list on the right where you select a port. The Statistics pane contains a table with a summary of event statistics and two drop-down lists where you can select a successful or failed event to view more information about that event. In the lower right of the window is a button to click to suspend or resume a channel and a home button to click to return to the iBSE Monitoring Settings window.



Channels: TestChan
Ports: all

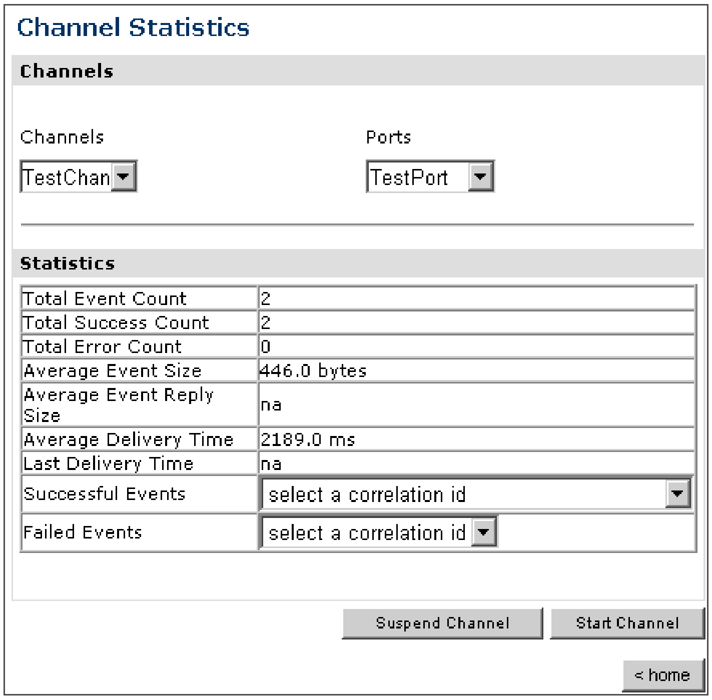
Total Event Count	3
Total Success Count	2
Total Error Count	1
Average Event Size	401.0 bytes
Average Event Reply Size	na
Average Delivery Time	1542.0 ms
Last Delivery Time	250 ms
Successful Events	select a correlation id
Failed Events	select a correlation id

a. To stop a channel at any time, click *Suspend Channel*.

b. To start the channel, click *Start Channel*.

5. From the Ports drop-down list, select a port for the channel.

The Port Level Event Summary (Channel Statistics) window opens as shown in the following image. The Channels pane contains a drop-down list on the left where you select a channel and a drop-down list on the right where you select a port. The Statistics pane contains a table with a summary of event statistics and two drop-down lists where you can select a successful or failed event to view more information about that event. In the lower right of the window is a button to click to suspend or resume a channel and a home button to click to return to the iBSE Monitoring Settings window.



Channel Statistics

Channels

Channels: TestChan

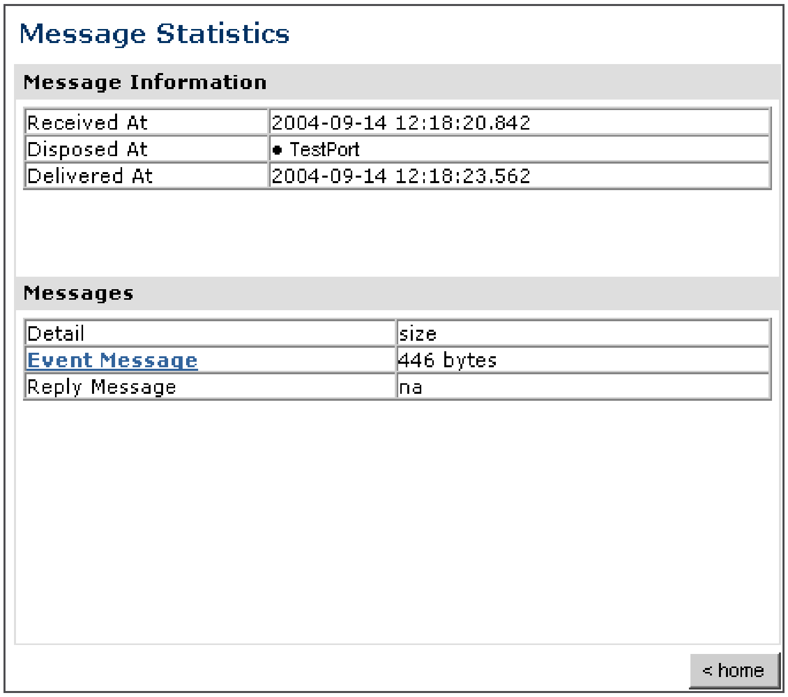
Ports: TestPort

Statistics

Total Event Count	2
Total Success Count	2
Total Error Count	0
Average Event Size	446.0 bytes
Average Event Reply Size	na
Average Delivery Time	2189.0 ms
Last Delivery Time	na
Successful Events	select a correlation id
Failed Events	select a correlation id

6. For more information about a successful event and its port, select an event based on its correlation ID from the Successful Events drop-down list.

   The Event Level Statistics (Message Statistics) window opens as shown in the following image. The Message Information pane contains a table of information pertaining to the event message. The Messages pane contains a table that shows the size of the event and reply messages.



Message Statistics

Message Information

Received At	2004-09-14 12:18:20.842
Disposed At	● TestPort
Delivered At	2004-09-14 12:18:23.562

Messages

Detail	size
Event Message	446 bytes
Reply Message	na

   a. To view the XML event document in your Web browser, click *Event Message*.

   b. To return to the iBSE Monitoring Settings window, click *home*.

Managing and Monitoring Services and Events Using the JCA Test Tool

The JCA Test Tool, which is also known as the JCA Installation Verification Program (IVP), provides a console to manage and monitor services and events currently in use. The console also displays resource usage and invocation statistics. These indicators can help you adjust your environment for optimum efficiency.

Procedure: How to Manage and Monitor Services Using the JCA Test Tool

To manage and monitor services using the JCA Test Tool:

1. Open a Web browser to:

   `http://`*`hostname`*`:`*`port`*`/iwjcaivp`

   where:

   `hostname`

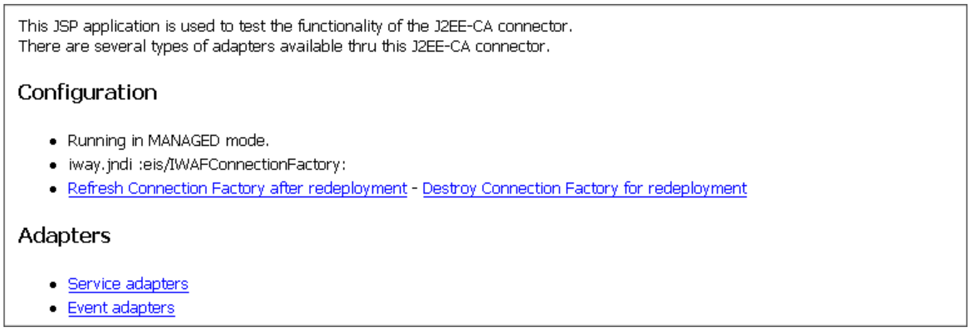
   Is the name of the machine where your application server is running.

   `port`

   Is the port for the domain you are using. The port for the default domain is 7001, for example:

   `http://localhost:7001/iwjcaivp`

   The following image shows the JCA Test Tool pane that opens. The pane contains a description of the function of the tool and configuration information, including options to change your connection settings. It also provides options for viewing service or event adapters.



This JSP application is used to test the functionality of the J2EE-CA connector.
There are several types of adapters available thru this J2EE-CA connector.

Configuration

- Running in MANAGED mode.
- iway.jndi :eis/IWAFConnectionFactory:
- Refresh Connection Factory after redeployment - Destroy Connection Factory for redeployment

Adapters

- Service adapters
- Event adapters

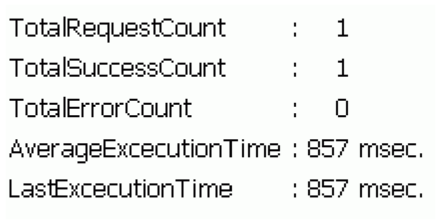
The JCA Test Tool runs in managed mode by default.

2. Perform the following steps to monitor the latest service adapter configuration.

   Note: You must perform these steps for every new adapter target that is created using a JCA implementation of Application Explorer. In addition, you also must perform these steps for every new JCA configuration that is created using Application Explorer.

   a. Click *Destroy Connection Factory for redeployment*.
   b. Redeploy the JCA connector.
   c. In the JCA Test Tool, click *Refresh Connection Factory after redeployment*.
3. Click *Service adapters*.
4. Select a service adapter to monitor.
   a. Click the desired target for your service adapter.
   b. In the Request area, enter a user name and password.
   c. In the Input Doc area, enter a request document created from the request schema for your service.
5. Click *Send*.

   The following image shows the updated statistics that appear for your service if the request is successful. The statistics include the total number of requests, successes, and errors and the average and last execution time in milliseconds.



```
TotalRequestCount      :   1
TotalSuccessCount      :   1
TotalErrorCount        :   0
AverageExcecutionTime : 857 msec.
LastExcecutionTime     : 857 msec.
```

Procedure: How to Manage and Monitor Events Using the JCA Test Tool

To manage and monitor events using the JCA Test Tool:

1. Open a Web browser to:

   `http://`*`hostname`*`:`*`port`*`/iwjcaivp`

   where:

   `hostname`

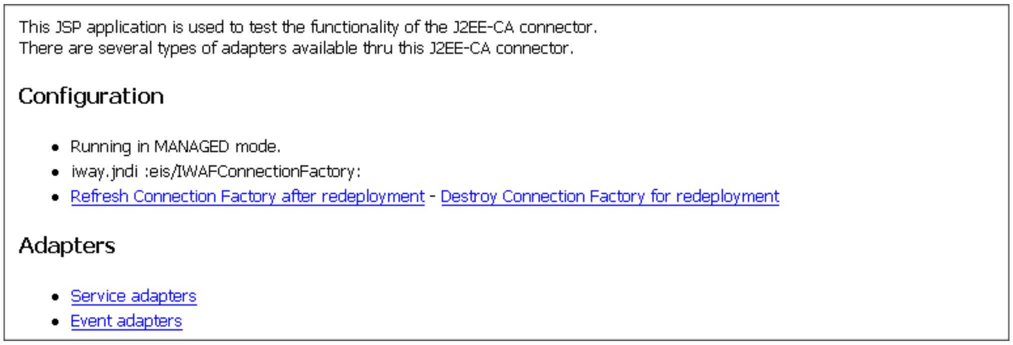
   Is the name of the machine where your application server is running.

   `port`

   Is the port for the domain you are using. The port for the default domain is 7001, for example:

   `http://localhost:7001/iwjcaivp`

   The JCA Test Tool pane opens as shown in the following image. The pane contains a description of the function of the tool and configuration information, including options to change your connection settings. It also provides options for viewing service or event adapters.



This JSP application is used to test the functionality of the J2EE-CA connector.
There are several types of adapters available thru this J2EE-CA connector.

Configuration

- Running in MANAGED mode.
- iway.jndi :eis/IWAFConnectionFactory:
- Refresh Connection Factory after redeployment - Destroy Connection Factory for redeployment

Adapters

- Service adapters
- Event adapters

   The JCA Test Tool runs in managed mode by default.

2. To monitor the latest event adapter configuration, perform the following steps.

   Note: You must perform these steps for every new adapter target that is created using a JCA implementation of Application Explorer. In addition, you must also perform these steps for every new JCA configuration that is created using Application Explorer.

   a. Click *Destroy Connection Factory for redeployment*.

   b. Redeploy the JCA connector.

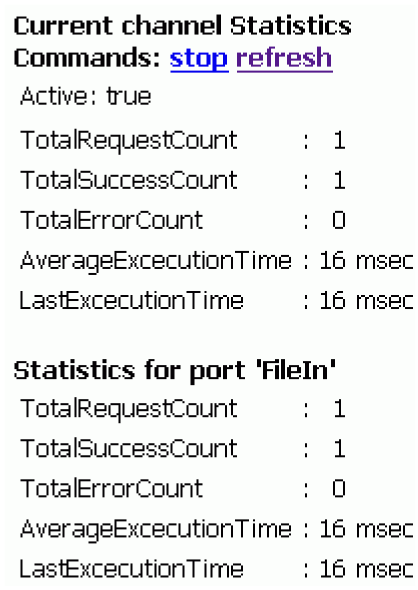
   c. In the JCA Test Tool, click *Refresh Connection Factory after redeployment*.

3. Click *Event adapters*.

   The Event Adapters pane opens.

4. Select the event adapter to monitor.
5. Click the desired channel for your event adapter.
6. Click *start*.

   The following image shows the updated statistics for your channel and the port. The statistics include the total number of requests, successes, and errors and the average and last execution time in milliseconds. The upper right of the pane contains options to start or refresh the channel.



```
Current channel Statistics
Commands: stop refresh
Active: true
TotalRequestCount       :  1
TotalSuccessCount       :  1
TotalErrorCount         :  0
AverageExcecutionTime : 16 msec
LastExcecutionTime      : 16 msec

Statistics for port 'FileIn'
TotalRequestCount       :  1
TotalSuccessCount       :  1
TotalErrorCount         :  0
AverageExcecutionTime : 16 msec
LastExcecutionTime      : 16 msec
```

Setting Engine Log Levels

The following procedures describe how to set engine log levels. For more information, see the *BEA WebLogic ERP Adapter Installation and Configuration* manual.

Procedure: How to Enable Tracing for Servlet iBSE

To enable tracing for Servlet iBSE:

1. Open the Servlet iBSE configuration window at:

   `http://hostname:port/ibse/IBSEConfig`

   where:

   `hostname`

   Is the name of the machine where your application server is running.

   `port`

   Is the port for the domain you are using. The port for the default domain is 7001, for example:

   `http://localhost:7001/ibse/IBSEConfig`

2. In the System pane, from the Debug drop-down list, select the level of tracing.
3. Click *Save*.

The default location for the trace information on Windows is:

`C:\Program FIles\bea\ibse\ibselogs`

Procedure: How to Enable Tracing for JCA

To enable tracing for JCA:

1. Extract the ra.xml file from the iwafjca.rar file.
2. Open the extracted ra.xml file in a text editor.
3. Locate and change the following setting:

   LogLevel. This setting can be set to DEBUG, INFO, or ERROR.

   ```
   <context-param>
   <config-property>
       <config-property-name>LogLevel</config-property-name>
       <config-property-type>java.lang.String</config-property-type>
       <config-property-value></config-property-value>
   </config-property>
   ```

   For example:

   ```
   <config-property-value>DEBUG</config-property-value>
   ```

   Leave the remainder of the file unchanged.
4. Save the file and exit the editor.
5. Add the file back to the iwafjca.rar file.
6. Redeploy the connector.

A directory in the configuration directory contains the logs. For example, on Windows the default location is the following:

```
C:\Program Files\iway55\config\base\log
```

The log files are named jca_*.log.

You should also review the logs generated by your application server.

Configuring Connection Pool Sizes

The following topic describes how to configure connection pool sizes for the Connector for JCA.

Procedure: How to Configure Connection Pool Sizes

To configure connection pool sizes:

1. Extract the ra.xml file from the iwafjca.rar file.
2. Open the extracted ra.xml file in a text editor.
3. Locate and change the following setting:

   pool-params. The JCA Resource Connector has an initial capacity value of 0 by default and cannot be changed. The maximum capacity value is 10 by default and can be changed to a higher value.

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE weblogic-connection-factory-dd (View Source for full
doctype...)>
- <weblogic-connection-factory-dd>
  <connection-factory-name>IWAFJCA</connection-factory-name>
  <jndi-name>eis/IWAFConnectionFactory</jndi-name>
  - <pool-params>
    <initial-capacity>0</initial-capacity>
    <max-capacity>10</max-capacity>
    <capacity-increment>1</capacity-increment>
    <shrinking-enabled>false</shrinking-enabled>
    <shrink-period-minutes>200</shrink-period-minutes>
  </pool-params>
  <security-principal-map />
</weblogic-connection-factory-dd>
```

4. Save the file and exit the editor.
5. Return the ra.xml file to the iwafjca.rar file.
6. Redeploy the connector.

Migrating Repositories

During design time, a repository is used to store metadata that is created when using Application Explorer to:

- Configure adapter connections.
- Browse EIS objects.
- Configure services.
- Configure listeners to listen for EIS events.

For more information on configuring repositories, see the *BEA WebLogic ERP Adapter Installation and Configuration* manual.

The information in the repository also is referenced at run time. For management purposes, you can migrate iBSE and JCA repositories to new destinations without affecting your existing configuration. For example, you can migrate a repository from a development environment to a production environment. The BEA WebLogic Server must be restarted to detect new repository changes.

File Repositories

To migrate a File repository to another destination, you copy the ibserepo.xml file from the following path:

```
drive:\Program Files\iWay55\bea\ibse\ibserepo.xml
```

where:

`drive`

Is the location of your BEA WebLogic ERP Adapter installation.

You can place the ibserepo.xml file in a new location that is a root directory of the iBSE Web application, for example:

```
drive:\ProductionConfig\bea\ibse\ibserepo.xml
```

iBSE Repositories

The following topic describes how to migrate an iBSE repository that is configured for Oracle. You can follow the same procedure if you want to migrate an iBSE repository that is configured for Microsoft SQL Server 2000, Sybase, or DB2. However, when you are configuring a new environment, you must execute the script that creates the repository tables for your database. In addition, verify that all required files and drivers for your database are in the class path. For more information on configuring repositories, see the *BEA WebLogic ERP Adapter Installation and Configuration* manual.

Note: The following procedure allows you to migrate only Web services. If migrating event handling information is one of your requirements, you must migrate at the database level. For more information, see *Migrating Event Handling Configurations* on page 1-1.

Procedure: How to Migrate an iBSE Repository Configured for Oracle

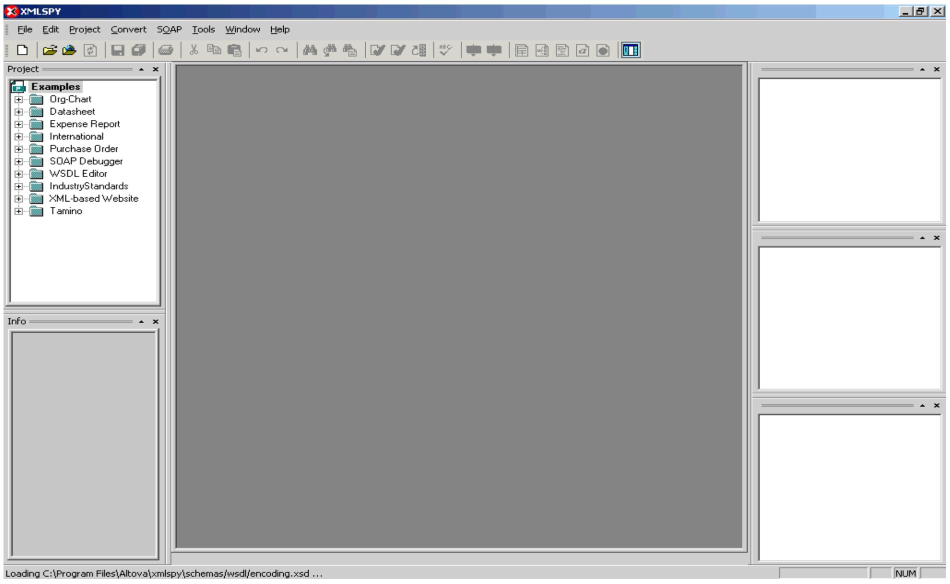
To migrate an iBSE repository that is configured for Oracle:

1. Copy the iBSE configuration service URL, for example:

   `http://localhost:7777/ibse/IBSEServlet/admin/iwconfig.ibs?wsdl`

2. Open a third party XML editor, for example, XMLSPY.

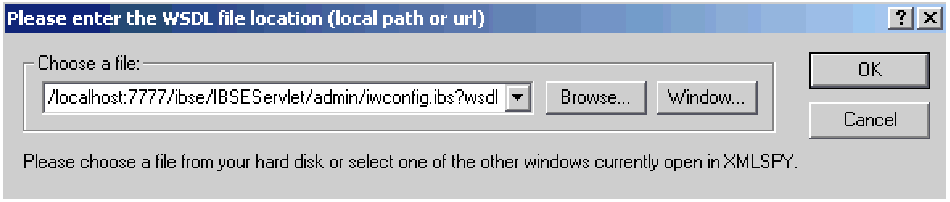
   The following image shows the XMLSPY window. The upper left has a Project pane that contains a tree of sample files, and the lower left has a blank Info pane. The middle pane is blank. The right side is divided into three blank panes.



Loading C:\Program Files\Altova\xmlspy\schemas\wsdl/encoding.xsd ...

3. From the SOAP menu, select *Create new SOAP request*.

   The WSDL file location dialog box opens as shown in the following image, where you enter a local path or URL. The dialog includes Browse, Window, OK, and Cancel buttons.

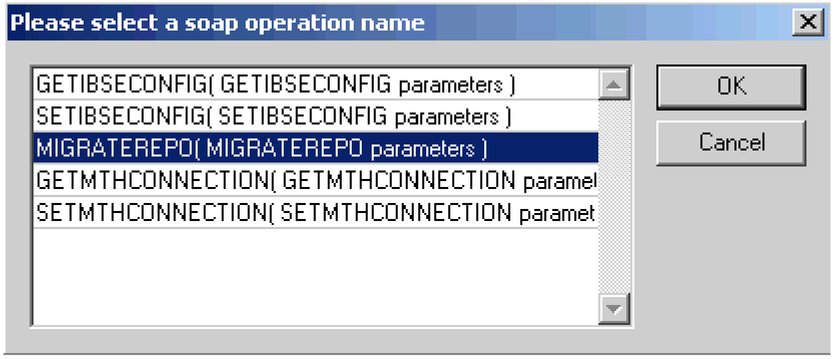

Please enter the WSDL file location (local path or url)

Choose a file: /localhost:7777/ibse/IBSEServlet/admin/iwconfig.ibs?wsdl

Please choose a file from your hard disk or select one of the other windows currently open in XMLSPY.

4. In the Choose a file field, paste the iBSE configuration service URL.
5. Click *OK*.

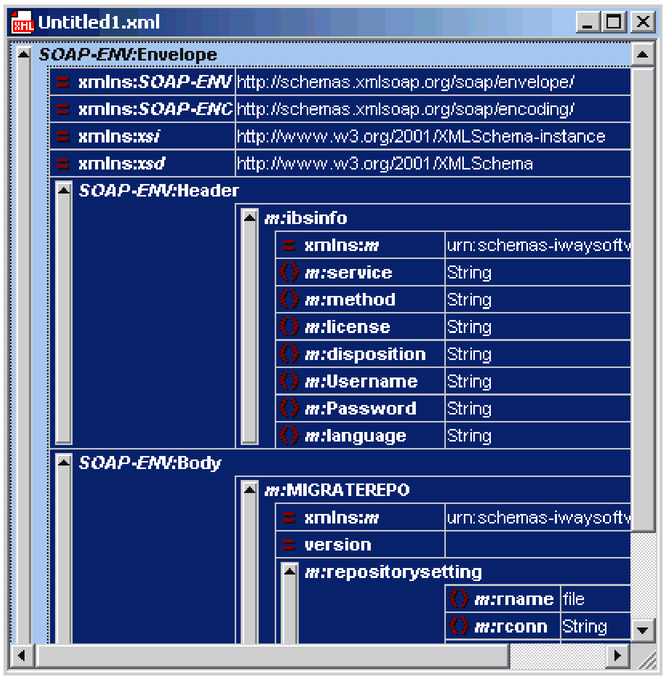
   The following image shows the soap operation name dialog box that opens with a list of available control methods.


Please select a soap operation name

GETIBSECONFIG(GETIBSECONFIG parameters)
SETIBSECONFIG(SETIBSECONFIG parameters)
MIGRATEREPO(MIGRATEREPO parameters)
GETMTHCONNECTION(GETMTHCONNECTION paramet
SETMTHCONNECTION(SETMTHCONNECTION paramet

6. Select the *MIGRATEREPO(MIGRATEREPO parameters)* control method and click *OK*.

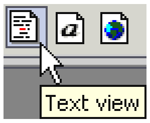
   The following image shows a portion of the window that opens with the structure of the SOAP envelope. It includes information about location and schemas.



```
Untitled1.xml
SOAP-ENV:Envelope
  xmlns:SOAP-ENV   http://schemas.xmlsoap.org/soap/envelope/
  xmlns:SOAP-ENC   http://schemas.xmlsoap.org/soap/encoding/
  xmlns:xsi        http://www.w3.org/2001/XMLSchema-instance
  xmlns:xsd        http://www.w3.org/2001/XMLSchema
  SOAP-ENV:Header
    m:ibsinfo
      xmlns:m          urn:schemas-iwaysoftv
      m:service        String
      m:method         String
      m:license        String
      m:disposition    String
      m:Username       String
      m:Password       String
      m:language       String
  SOAP-ENV:Body
    m:MIGRATEREPO
      xmlns:m          urn:schemas-iwaysoftv
      version
      m:repositorysetting
        m:rname        file
        m:rconn        String
```

7. Locate the *Text view* icon in the tool bar.

   In the following image, the pointer points to the Text view icon.



8. To display the structure of the SOAP envelope as text, click the *Text view* icon.

   The <SOAP-ENV:Header> tag is not required and can be deleted from the SOAP envelope.

9. Locate the following section:

```
<m:MIGRATEREPO
xmlns:m="urn:schemas-iwaysoftware-com:jul2003:ibse:config" version="">
<m:repositorysetting>
<m:rname>oracle</m:rname>
<m:rconn>String</m:rconn>
<m:rdriver>String</m:rdriver>
<m:ruser>String</m:ruser>
<m:rpwd>String</m:rpwd>
</m:repositorysetting>
<m:servicename>String</m:servicename>
</m:MIGRATEREPO>
```

   a. For the <m:rconn> tag, replace the String placeholder with the repository URL where you want to migrate your existing iBSE repository.

      For example, the Oracle repository URL has the following format:

      ```
      jdbc:oracle:thin:@[host]:[port]:[sid]
      ```

   b. For the <m:rdriver> tag, replace the String placeholder with the location of your Oracle driver.

      Note: This is an optional tag. If you do not specify a value, the default Oracle JDBC driver is used.

   c. For the <m:ruser> tag, replace the String placeholder with a valid user name to access the Oracle repository.

   d. For the <m:rpwd> tag, replace the String placeholder with a valid password to access the Oracle repository.

10. Perform one of the following migration options.

    If you want to migrate a **single** Web service from the current iBSE repository, enter the Web service name in the <m:servicename> tag, for example:

    ```
    <m:servicename>Service1</m:servicename>
    ```

    If you want to migrate **multiple** Web services from the current iBSE repository, duplicate the <m:servicename> tag for each Web service, for example:

    ```
    <m:servicename>Service1</m:servicename>
    <m:servicename>Service2</m:servicename>
    ```

    If you want to migrate **all** Web services from the current iBSE repository, remove the <m:servicename> tag.

11. From the SOAP menu, select *Send request to server.*

    Your iBSE repository and the Web services you specified migrate to the new Oracle repository URL that you specified.

JCA Repositories

The following procedure describes how to migrate a JCA repository. For more information on configuring JCA repositories, see the *BEA WebLogic ERP Adapter Installation and Configuration* manual.

Procedure: How to Migrate a JCA Repository

To migrate a JCA repository:

1. Navigate to the location of your JCA configuration directory where the repository schemas and other information is stored, for example:

   ```
   C:\Program Files\iway55\config\base
   ```

2. Locate and copy the *repository.xml* file.
3. Place this file in a new JCA configuration directory to migrate the existing repository.

   Your JCA repository migrates to the new JCA configuration directory.

Migrating Event Handling Configurations

This topic describes how to migrate your iBSE repositories at a database level for Microsoft SQL Server 2000, Oracle, Sybase, or DB2. You can use this information to migrate event handling information, for example, port or channel configurations.

Procedure: How to Migrate a Microsoft SQL Server 2000 Repository

To migrate a Microsoft SQL Server 2000 repository:

1. Open a command prompt and navigate to the setup directory.

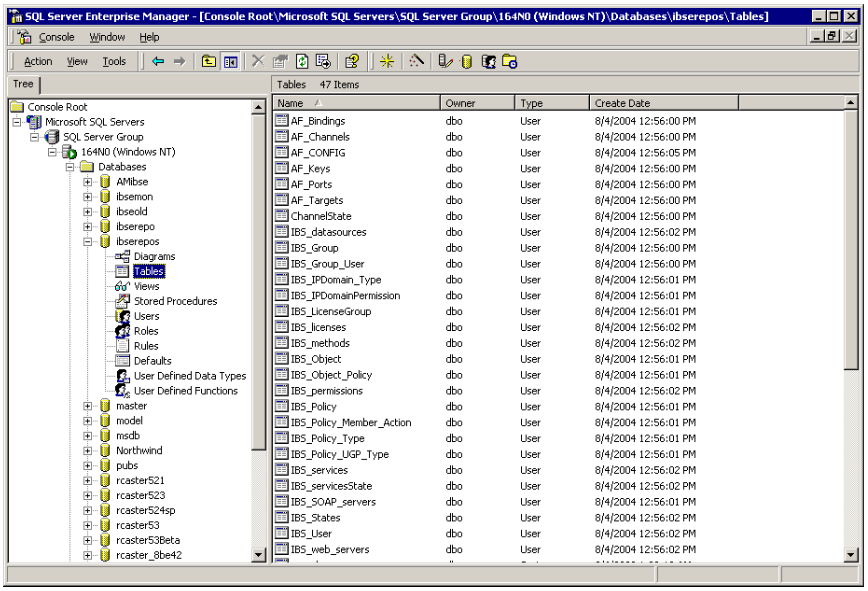
   The default location on Windows is:

   ```
   C:\Program Files\iWay55\etc\setup
   ```

   This directory contains SQL to create the repository tables in the following file:

   ```
   iwse.sql
   ```

You can use iwse.sql to create the database tables that are used by iBSE. For example, the following image shows the tree in the left pane and tables in the right pane. The tables are listed by name in one column with corresponding columns for information about owner, type, and the date the table was created.



SQL Server Enterprise Manager - [Console Root\Microsoft SQL Servers\SQL Server Group\164N0 (Windows NT)\Databases\ibserepos\Tables]

Tables 47 Items

Name	Owner	Type	Create Date
AF_Bindings	dbo	User	8/4/2004 12:56:00 PM
AF_Channels	dbo	User	8/4/2004 12:56:00 PM
AF_CONFIG	dbo	User	8/4/2004 12:56:05 PM
AF_Keys	dbo	User	8/4/2004 12:56:00 PM
AF_Ports	dbo	User	8/4/2004 12:56:00 PM
AF_Targets	dbo	User	8/4/2004 12:56:00 PM
ChannelState	dbo	User	8/4/2004 12:56:00 PM
IBS_datasources	dbo	User	8/4/2004 12:56:02 PM
IBS_Group	dbo	User	8/4/2004 12:56:00 PM
IBS_Group_User	dbo	User	8/4/2004 12:56:00 PM
IBS_IPDomain_Type	dbo	User	8/4/2004 12:56:01 PM
IBS_IPDomainPermission	dbo	User	8/4/2004 12:56:01 PM
IBS_LicenseGroup	dbo	User	8/4/2004 12:56:01 PM
IBS_licenses	dbo	User	8/4/2004 12:56:02 PM
IBS_methods	dbo	User	8/4/2004 12:56:02 PM
IBS_Object	dbo	User	8/4/2004 12:56:01 PM
IBS_Object_Policy	dbo	User	8/4/2004 12:56:01 PM
IBS_permissions	dbo	User	8/4/2004 12:56:02 PM
IBS_Policy	dbo	User	8/4/2004 12:56:01 PM
IBS_Policy_Member_Action	dbo	User	8/4/2004 12:56:01 PM
IBS_Policy_Type	dbo	User	8/4/2004 12:56:01 PM
IBS_Policy_UGP_Type	dbo	User	8/4/2004 12:56:01 PM
IBS_services	dbo	User	8/4/2004 12:56:02 PM
IBS_servicesState	dbo	User	8/4/2004 12:56:02 PM
IBS_SOAP_servers	dbo	User	8/4/2004 12:56:01 PM
IBS_States	dbo	User	8/4/2004 12:56:02 PM
IBS_User	dbo	User	8/4/2004 12:56:02 PM
IBS_web_servers	dbo	User	8/4/2004 12:56:02 PM

For more information on configuring the Microsoft SQL Server 2000 repository, see the *BEA WebLogic ERP Adapter Installation and Configuration* manual.

2. To migrate the tables that were created by the iwse.sql script for iBSE, use your Microsoft SQL Server 2000 database tool set. For more information, consult your database administrator.

Procedure: How to Migrate an Oracle Repository

To migrate an Oracle repository:

1. Open a command prompt and navigate to the setup directory.

   The default location on Windows is:

   `C:\Program Files\iWay55\etc\setup`

   This directory contains SQL to create the repository tables in the following files:

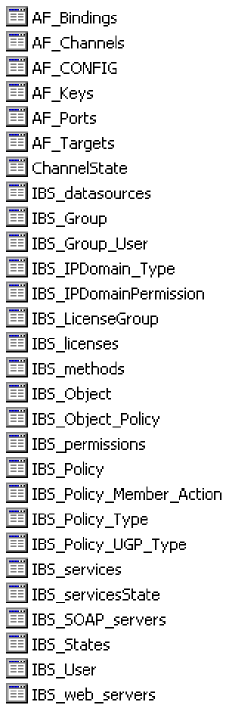
   For Oracle 8:

   `iwse.ora`

   For Oracle 9:

   `iwse.ora9`

2. To create the Oracle database tables that are used by iBSE, use the SQL script as shown in the example in the following image that shows a list of tables.



AF_Bindings
AF_Channels
AF_CONFIG
AF_Keys
AF_Ports
AF_Targets
ChannelState
IBS_datasources
IBS_Group
IBS_Group_User
IBS_IPDomain_Type
IBS_IPDomainPermission
IBS_LicenseGroup
IBS_licenses
IBS_methods
IBS_Object
IBS_Object_Policy
IBS_permissions
IBS_Policy
IBS_Policy_Member_Action
IBS_Policy_Type
IBS_Policy_UGP_Type
IBS_services
IBS_servicesState
IBS_SOAP_servers
IBS_States
IBS_User
IBS_web_servers

   For more information on configuring the Oracle repository, see the *BEA WebLogic ERP Adapter Installation and Configuration* manual.

3. To migrate the tables that were created by the SQL script for iBSE, use your Oracle database tool set. For more information, consult your database administrator.

Procedure: How to Migrate a Sybase Repository

To migrate a Sybase repository:

1. Open a command prompt and navigate to the setup directory.

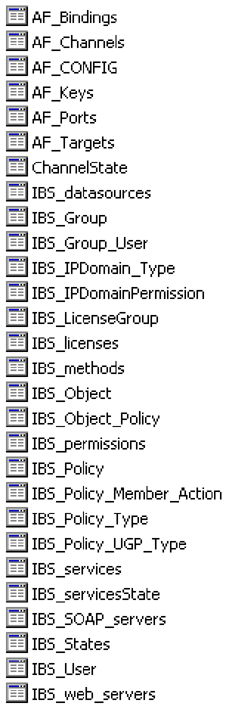
   The default location on Windows is:

   `C:\Program Files\iWay55\etc\setup`

   This directory contains SQL to create the repository tables in the following file:

   `sybase-iwse.sql`

2. To create the Sybase database tables that are used by iBSE, use the SQL script as shown in the example in the following image that shows a list of tables.

   AF_Bindings
   AF_Channels
   AF_CONFIG
   AF_Keys
   AF_Ports
   AF_Targets
   ChannelState
   IBS_datasources
   IBS_Group
   IBS_Group_User
   IBS_IPDomain_Type
   IBS_IPDomainPermission
   IBS_LicenseGroup
   IBS_licenses
   IBS_methods
   IBS_Object
   IBS_Object_Policy
   IBS_permissions
   IBS_Policy
   IBS_Policy_Member_Action
   IBS_Policy_Type
   IBS_Policy_UGP_Type
   IBS_services
   IBS_servicesState
   IBS_SOAP_servers
   IBS_States
   IBS_User
   IBS_web_servers

   For more information on configuring the Sybase repository, see the *BEA WebLogic ERP Adapter Installation and Configuration* manual.

3. To migrate the tables that were created by the SQL script for iBSE, use your Sybase database tool set. For more information, consult your database administrator.

Procedure: How to Migrate a DB2 Repository

To migrate a DB2 repository:

1. Open a command prompt and navigate to the setup directory.

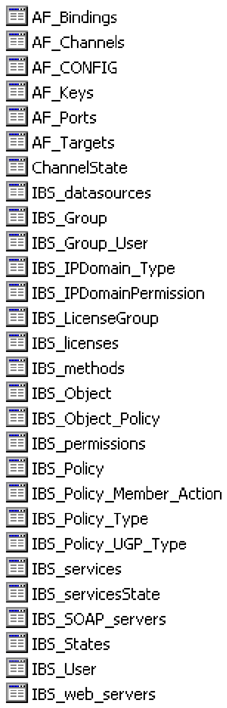
   The default location on Windows is:

   `C:\Program Files\iWay55\etc\setup`

   This directory contains SQL to create the repository tables in the following file:

   `db2-iwse.sql`

2. To create the DB2 database tables that are used by iBSE, use the SQL script as shown in the example in the following image that shows a list of tables.



AF_Bindings
AF_Channels
AF_CONFIG
AF_Keys
AF_Ports
AF_Targets
ChannelState
IBS_datasources
IBS_Group
IBS_Group_User
IBS_IPDomain_Type
IBS_IPDomainPermission
IBS_LicenseGroup
IBS_licenses
IBS_methods
IBS_Object
IBS_Object_Policy
IBS_permissions
IBS_Policy
IBS_Policy_Member_Action
IBS_Policy_Type
IBS_Policy_UGP_Type
IBS_services
IBS_servicesState
IBS_SOAP_servers
IBS_States
IBS_User
IBS_web_servers

For more information on configuring the DB2 repository, see the *BEA WebLogic ERP Adapter Installation and Configuration* manual.

You can migrate the tables that were created by the SQL script for iBSE using your DB2 database toolset. For more information, consult your database administrator.

Exporting or Importing Targets

After you migrate your repository, you can export or import targets with their connection information and persistent data between repositories.

Procedure: How to Export a Target

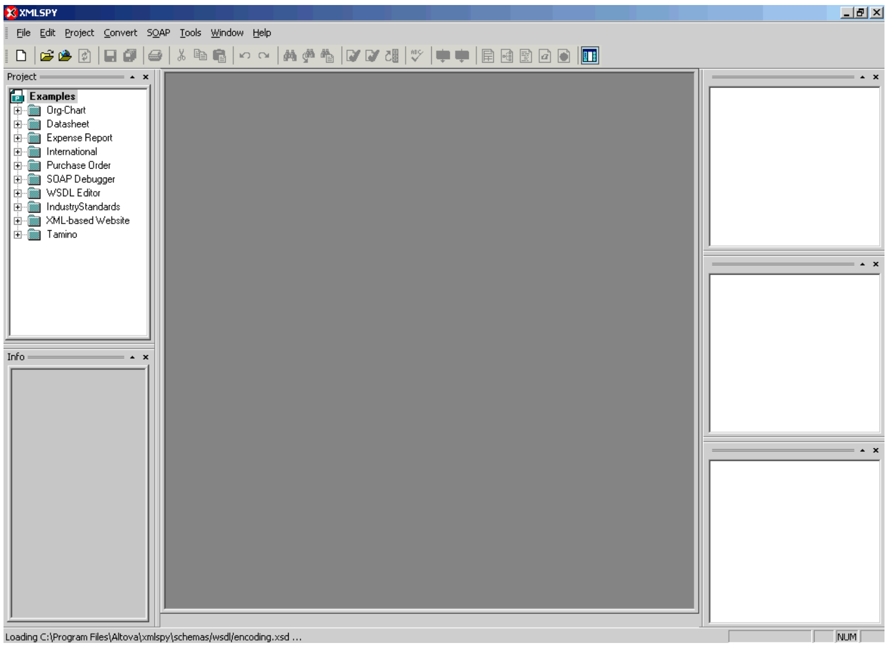
To export a target:

1. Copy the iBSE administrative services for Application Explorer URL, for example:

   `http://localhost:7777/ibse/IBSEServlet/admin/iwae.ibs?wsdl`

2. Open a third party XML editor, for example, XMLSPY.

   The following image shows the XMLSPY window. The upper left has a Project pane that contains a tree of sample files, and the lower left has a blank Info pane. The middle pane is blank. The right side is divided into three blank panes.



Loading C:\Program Files\Altova\xmlspy\schemas\wsdl\encoding.xsd ...

3. From the SOAP menu, select *Create new SOAP request*.

   The WSDL file location dialog box opens.

4. In the Choose a file field, paste the iBSE administrative services for Application Explorer URL.

5. Click *OK*.

   The soap operation name dialog box opens and lists the available control methods.

6. Select the *EXPORTTARGET(EXPORTTARGET parameters)* control method and click *OK*.

   A window opens that shows the structure of the SOAP envelope.

7. Locate the *Text view* icon in the tool bar.

8. To display the structure of the SOAP envelope as text, click the *Text view* icon.

   The <SOAP-ENV:Header> tag is not required and can be deleted from the SOAP envelope.

9. Locate the following section:

   ```
   <m:EXPORTTARGET
   xmlns:m="urn:schemas-iwaysoftware-com:dec2002:iwse:af">
   <m:target>String</m:target>
   <m:name>String</m:name>
   </m:EXPORTTARGET>
   ```

   a. For the <m:target> tag, replace the String placeholder with the EIS target system name as it appears in Application Explorer and verify whether this value is case sensitive.

   b. For the <m:name> tag, replace the String placeholder with the name of the target you want to export.

10. From the SOAP menu, select *Send request to server.*

    A response is returned that contains the <m: exporttime> and <m: contents> elements. You must use these elements when importing your target.

Procedure: How to Import a Target

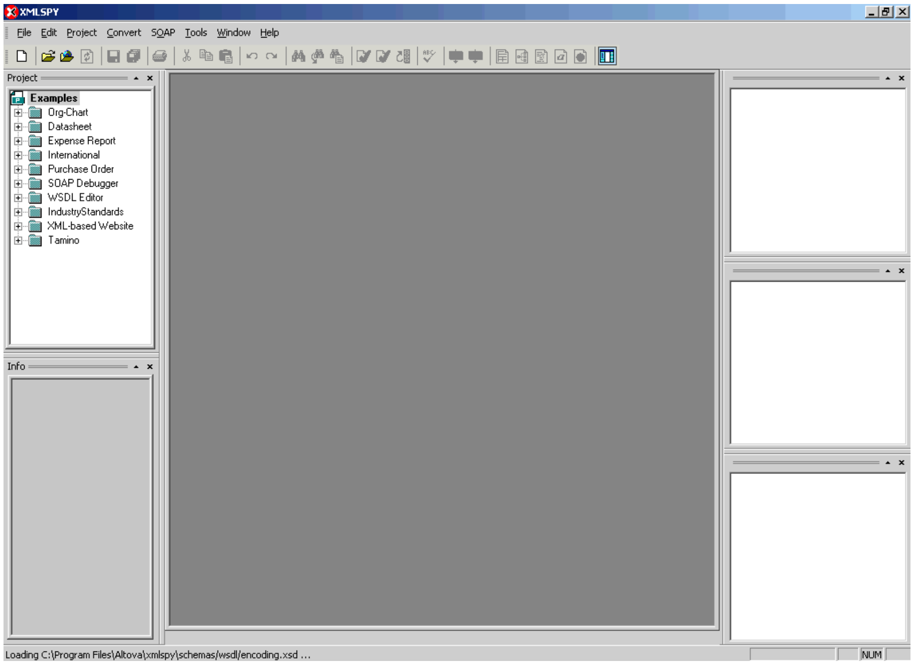
To import a target:

1. Copy the iBSE administrative services for Application Explorer URL, for example:

   `http://localhost:7777/ibse/IBSEServlet/admin/iwae.ibs?wsdl`

2. Open a third party XML editor, for example, XMLSPY.

   The following image shows the XMLSPY window. The upper left has a Project pane that contains a tree of sample files, and the lower left has a blank Info pane. The middle pane is blank. The right side is divided into three blank panes.



Loading C:\Program Files\Altova\xmlspy\schemas\wsdl/encoding.xsd ...

3. From the SOAP menu, select *Create new SOAP request*.

   The WSDL file location dialog box opens.

4. In the Choose a file field, paste the iBSE administrative services for Application Explorer URL and click *OK*.

   The soap operation name dialog box opens and lists the available control methods.

5. Select the *IMPORTTARGET(IMPORTTARGET parameters)* control method and click *OK*.

   A window opens, which shows the structure of the SOAP envelope.

6. Locate the *Text view* icon in the tool bar.
7. To display the structure of the SOAP envelope as text, click the *Text view* icon.

   The <SOAP-ENV:Header> tag is not required and can be deleted from the SOAP envelope.
8. Locate the following section:

```
<m:IMPORTTARGET
xmlns:m="urn:schemas-iwaysoftware-com:dec2002:iwse:af">
<m:targetinstance>
<m:target>String</m:target>
<m:name>String</m:name>
<m:description>String</m:description>
<m:repositoryid>String</m:repositoryid>
<m:exporttime>2001-12-17T09:30:47-05:00</m:exporttime>
<m:contents>R0lGODlhcgGSALMAAAQCAEMmCZtuMFQxDS8b</m:contents>
</m:targetinstance>
</m:IMPORTTARGET>
```

   a. For the <m:target> tag, replace the String placeholder with the EIS target system name.
   b. For the <m:name> tag, replace the String placeholder with the new name of the target you want to import.
   c. For the <m:description> tag, replace the String placeholder with a description of the target.
   d. For the <m:repositoryid> tag, copy and paste the contents of the <m:repositoryid> tag that was returned when you exported your target.
   e. For the <m: exporttime> tag, copy and paste the contents of the <m: exporttime> tag that was returned when you exported your target.
   f. For the <m: contents> tag, copy and paste the contents of the <m: contents> tag that was returned when you exported your target.
9. From the SOAP menu, select *Send request to server.*

Retrieving or Updating Web Service Method Connection Information

After you migrate your repository, you can retrieve or update connection information for your Web service methods.

Procedure: How to Retrieve Web Service Method Connection Information

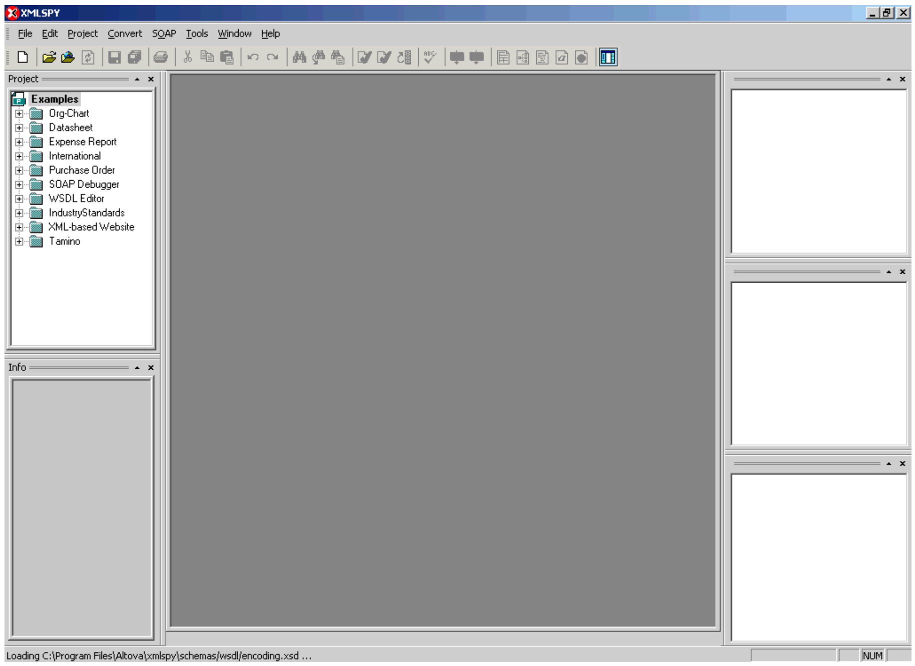
To retrieve Web service method connection information:

1. Copy the iBSE configuration service URL, for example:

   `http://localhost:7777/ibse/IBSEServlet/admin/iwconfig.ibs?wsdl`

2. Open a third party XML editor, for example, XMLSPY.

   The following image shows the XMLSPY window. The upper left has a Project pane that contains a tree of sample files, and the lower left has a blank Info pane. The middle pane is blank. The right side is divided into three blank panes.



Loading C:\Program Files\Altova\xmlspy\schemas/wsdl/encoding.xsd ...

3. From the SOAP menu, select *Create new SOAP request*.

   The WSDL file location dialog box opens.

4. In the Choose a file field, paste the iBSE configuration service URL, and click *OK*.

   The soap operation name dialog box opens and lists the available control methods.

5. Select the *GETMTHCONNECTION(GETMTHCONNECTION parameters)* control method and click *OK*.

   A window opens, which shows the structure of the SOAP envelope.

6. Locate the *Text view* icon in the tool bar.

7. To display the structure of the SOAP envelope as text, click the *Text view* icon.

   The <SOAP-ENV:Header> tag is not required and can be deleted from the SOAP envelope.

8. Locate the following section:

   ```
   <m:GETMTHCONNECTION
   xmlns:m="urn:schemas-iwaysoftware-com:jul2003:ibse:config">
   <m:servicename>String</m:servicename>
   <m:methodname>String</m:methodname>
   </m:GETMTHCONNECTION>
   ```

   a. For the <m:servicename> tag, replace the String placeholder with the name of the Web service.

   b. For the <m:methodname> tag, replace the String placeholder with name of the Web service method.

9. From the SOAP menu, select *Send request to server.*

   A response is returned that contains the <m: descriptor> element. You must use this element when updating your Web service method.

Procedure: How to Update Web Service Method Connection Information

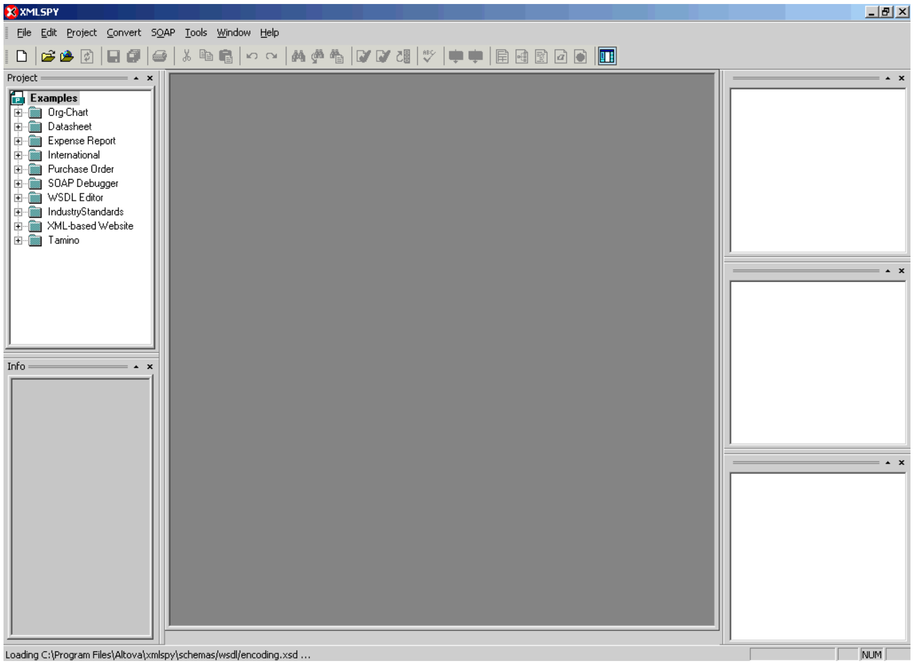
To update Web service method connection information:

1. Copy the iBSE configuration service URL, for example:

   `http://localhost:7777/ibse/IBSEServlet/admin/iwconfig.ibs?wsdl`

2. Open a third party XML editor, for example, XMLSPY.

   The following image shows the XMLSPY window. The upper left has a Project pane that contains a tree of sample files, and the lower left has a blank Info pane. The middle pane is blank. The right side is divided into three blank panes.



3. From the SOAP menu, select *Create new SOAP request*.

   The WSDL file location dialog box opens.

4. In the Choose a file field, paste the iBSE configuration service URL, and click *OK*.

   The soap operation name dialog box opens and lists the available control methods.

5. Select the *SETMTHCONNECTION(SETMTHCONNECTION parameters)* control method and click *OK*.

   A window opens that shows the structure of the SOAP envelope.

6. Locate the *Text view* icon in the tool bar.
7. To display the structure of the SOAP envelope as text, click the *Text view* icon.

   The <SOAP-ENV:Header> tag is not required and can be deleted from the SOAP envelope.
8. Locate the following section:

```
<m:SETMTHCONNECTION
xmlns:m="urn:schemas-iwaysoftware-com:jul2003:ibse:config">
<m:servicename>String</m:servicename>
<m:methodname>String</m:methodname>
<m:descriptor format="" channel="">
     <m:option title="">
       <m:group title="">
          <m:param/>
       </m:group>
     </m:option>
</m:descriptor>
</m:SETMTHCONNECTION>
```

   a. For the <m:servicename> tag, replace the String placeholder with the name of the Web service.
   b. For the <m:methodname> tag, replace the String placeholder with the name of the Web service method.
   c. For the <m: descriptor> tag, copy and paste the contents of the <m: descriptor> tag that was returned when you retrieved Web Service method connection information.
9. Modify the contents of the <m: descriptor> tag to change the existing Web Service method connection information.
10. From the SOAP menu, select *Send request to server.*

Starting or Stopping a Channel Programmatically

The following topic describes how to start or stop a channel programmatically.

Procedure: How to Start a Channel Programmatically

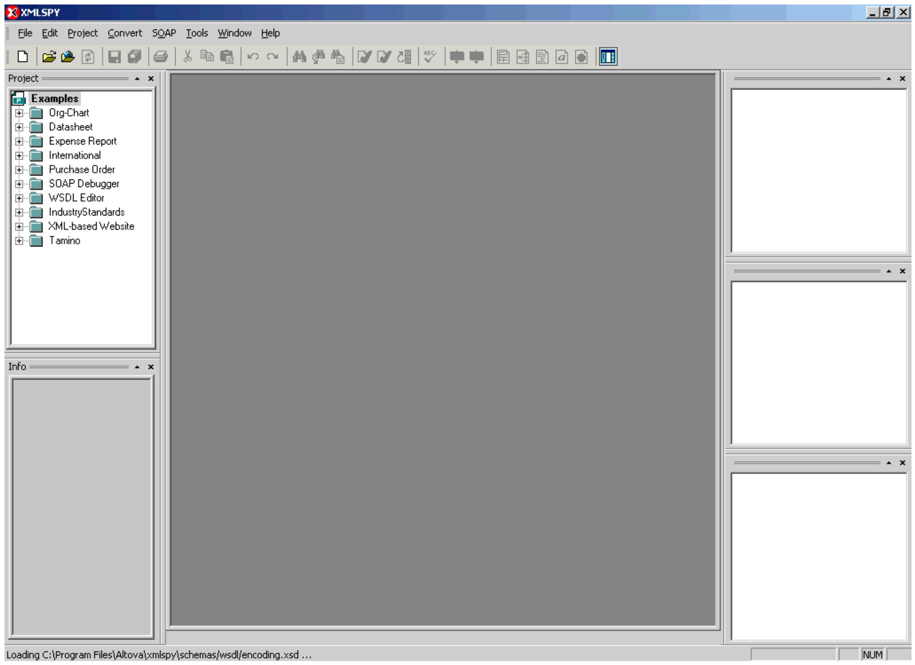
To start a channel programmatically:

1. Copy the iBSE control event URL, for example:

   `http://localhost:7777/ibse/IBSEServlet/admin/iwevent.ibs?wsdl`

2. Open a third party XML editor, for example, XMLSPY.

   The following image shows the XMLSPY window. The upper left has a Project pane that contains a tree of sample files, and the lower left has a blank Info pane. The middle pane is blank. The right side is divided into three blank panes.



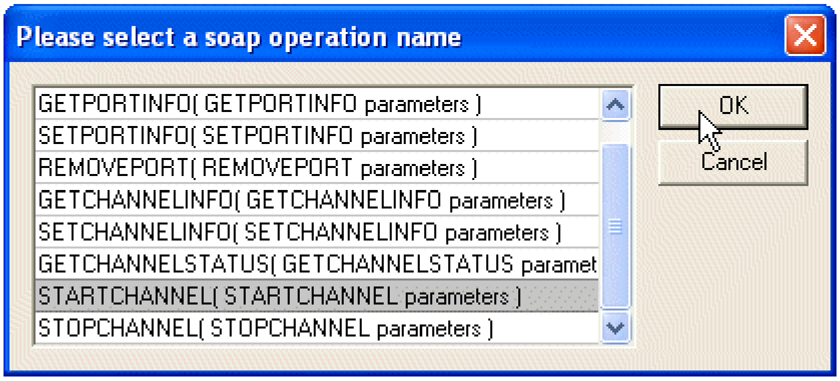
Loading C:\Program Files\Altova\xmlspy\schemas\wsdl\encoding.xsd ...

3. From the SOAP menu, select *Create new SOAP request*.

   The WSDL file location dialog box opens.

4. In the Choose a file field, paste the iBSE control event URL, and click *OK*.

The following image shows the soap operation name dialog box that opens with a list of available control methods.



Please select a soap operation name

GETPORTINFO(GETPORTINFO parameters)
SETPORTINFO(SETPORTINFO parameters)
REMOVEPORT(REMOVEPORT parameters)
GETCHANNELINFO(GETCHANNELINFO parameters)
SETCHANNELINFO(SETCHANNELINFO parameters)
GETCHANNELSTATUS(GETCHANNELSTATUS paramet
STARTCHANNEL(STARTCHANNEL parameters)
STOPCHANNEL(STOPCHANNEL parameters)

5. Select the *STARTCHANNEL(STARTCHANNEL parameters)* control method and click *OK*.

A window opens, which shows the structure of the SOAP envelope.

6. Locate the *Text view* icon in the tool bar.

7. To display the structure of the SOAP envelope as text, click the *Text view* icon.

The <SOAP-ENV:Header> tag is not required and can be deleted from the SOAP envelope.

8. Locate the following section:

```
<SOAP-ENV:Body>
  <m:STARTCHANNEL
     xmlns:m="urn:schemas-iwaysoftware-com:dec2002:iwse:event">
           <m:channel>String</m:channel>
  </m:STARTCHANNEL>
</SOAP-ENV:Body>
```

9. For the <m:channel> tag, replace the String placeholder with the name of the channel you want to start.

10. From the SOAP menu, select *Send request to server.*

Procedure: How to Stop a Channel Programmatically

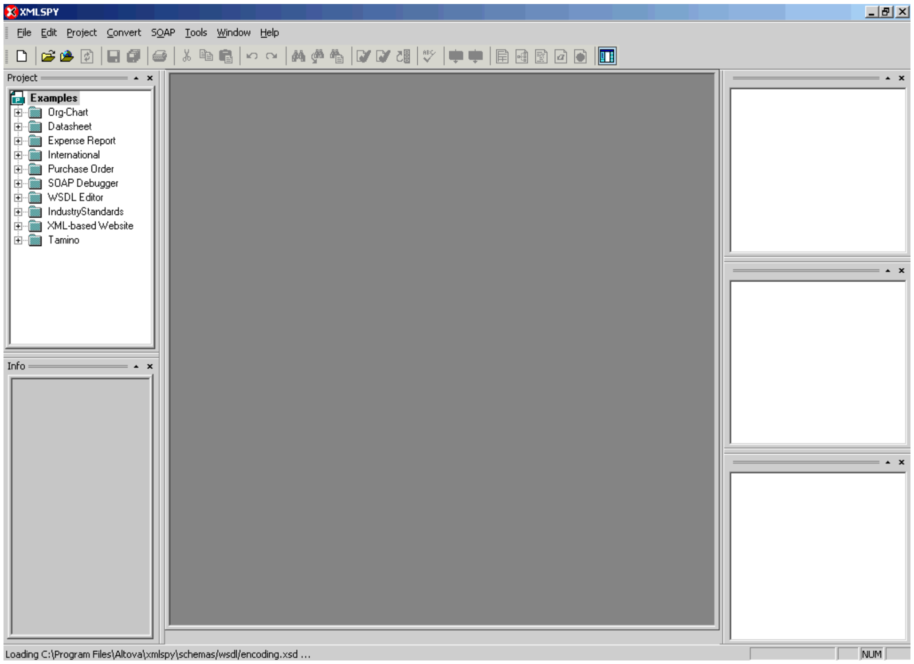
To stop a channel programmatically:

1. Copy the iBSE control event URL, for example:

   `http://localhost:7777/ibse/IBSEServlet/admin/iwevent.ibs?wsdl`

2. Open a third party XML editor, for example, XMLSPY.

   The following image shows the XMLSPY window. The upper left has a Project pane that contains a tree of sample files, and the lower left has a blank Info pane. The middle pane is blank. The right side is divided into three blank panes.



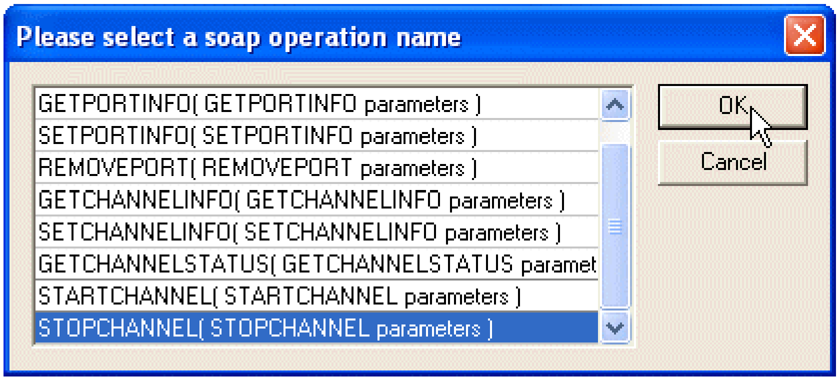
Loading C:\Program Files\Altova\xmlspy\schemas\wsdl/encoding.xsd ...

3. From the SOAP menu, select *Create new SOAP request*.

   The WSDL file location dialog box opens.

4. In the Choose a file field, paste the iBSE control event URL, and click *OK*.

   The following image shows the soap operation name dialog box that opens with a list of available control methods.



GETPORTINFO(GETPORTINFO parameters)
SETPORTINFO(SETPORTINFO parameters)
REMOVEPORT(REMOVEPORT parameters)
GETCHANNELINFO(GETCHANNELINFO parameters)
SETCHANNELINFO(SETCHANNELINFO parameters)
GETCHANNELSTATUS(GETCHANNELSTATUS paramet
STARTCHANNEL(STARTCHANNEL parameters)
STOPCHANNEL(STOPCHANNEL parameters)

5. Select the *STOPCHANNEL(STOPCHANNEL parameters)* control method and click *OK*.

   A window opens, which shows the structure of the SOAP envelope.

6. Locate the *Text view* icon in the tool bar.

7. To display the structure of the SOAP envelope as text, click the *Text view* icon.

   The <SOAP-ENV:Header> tag is not required and can be deleted from the SOAP envelope.

8. Locate the following section:

```
<SOAP-ENV:Body>
  <m:STOPCHANNEL
     xmlns:m="urn:schemas-iwaysoftware-com:dec2002:iwse:event">
           <m:channel>String</m:channel>
  </m:STOPCHANNEL>
</SOAP-ENV:Body>
```

9. For the <m:channel> tag, replace the String placeholder with the name of the channel you want to stop.

10. From the SOAP menu, select *Send request to Server.*

CHAPTER 8

Understanding SAP Events

Topics:

The following topics provide an overview of event functionality in SAP and describe how to configure and test your SAP system for event processing.

Overview

An event in SAP is defined as an occurrence of a status change in an object. The event is created when the relevant status change occurs. You or SAP must implement event creation.

An event is created from a specific application program (the event creator) and then "published" system-wide. An unlimited number of receivers can respond to the event with their own "response mechanisms". An event is usually defined as a component of an object type.

SAP "pseudo events" are events that are not processed by the SAP Event manager, but are called from an ABAP program or Remote Function call (using the Destination parameter).

Related Concepts and Terminology

The following topics list and define specific terminology related to SAP and SAP event handling.

Client and Server Programs

RFC (Remote Function Call) programs for non-SAP systems can function as either the caller or the called program in an RFC communication. The two types of RFC programs are:

- RFC Client
- RFC Server

The RFC client is the instance that calls the RFC to execute the function that is provided by an RFC server. The functions that can be executed remotely are called RFC functions, and the functions provided by the RFC API are called RFC calls.

SAP Gateway

The SAP Gateway is a secure BEA WebLogic Server. No connections are accepted unless they were pre-registered previously from the SAP presentation Client. A server connection presents itself to the Gateway and exposes a Program Identifier. If the Program Identifier is found in the list of registered Program IDs, the Gateway server then offers a connection to the server, which accepts a connection.

The Program ID then is linked with an RFC Destination within SAP, which enables SAP Function Modules and ALE documents (IDocs or BAPI IDocs) to be routed to the destination. The RFC Destination functions as a tag to mask the Program ID to SAP users.

An RFC server program can be registered with the SAP Gateway and wait for incoming RFC call requests. An RFC server program registers itself under a Program ID at an SAP Gateway and not for a specific SAP system.

In SAPGUI, the destination must be defined with transaction SM59, using connection type T and Register Mode. Moreover, this entry must contain information on the SAP Gateway where the RFC server program is registered.

Program IDs and Load Balancing

If the Gateway Server has a connection to a particular server instance and another server instance presents itself to the Gateway, the Gateway offers the connection and then begins functioning in Load Balancing mode. Using a proprietary algorithm, the Gateway sends different messages to each server depending on demand and total processing time. This could cause unpredictable results in the BEA WebLogic scenario where messages are validated by schema and application.

When configuring multiple events in BEA WebLogic using a single SAP program ID, SAP load balances the event data. For example, if multiple remote function calls or BAPIs use the same program ID (for example, IWAYID) and multiple SAP listeners are configured with this program ID, then SAP sends one request to one listener and the next to another listener, and so on.

The SAP Gateway Server includes a load balancing algorithm. This mechanism is proprietary to SAP application development and may work by comparing total throughput of the connection, the number of times in wait state, and so on. This means connection 1 may receive nine messages and connection 2 may receive one message. If five of nine messages are rejected for schema validation and the message on the other ID is rejected for schema validation, the customer can very easily make a case of missing messages.

Registering Your Program ID in SAPGUI

To enable your SAP system to issue the following calls or interfaces to the SAP event adapter, you must register your program ID under an RFC destination.

- Remote Function Calls (RFC)
- Business Application Programming Interfaces (BAPI)
- Intermediate Documents (IDoc)

The RFC destination is a symbolic name (for example, IWAYDEST) that is used to direct events to a target system, masking the program ID. The Program ID is configured in both SAPGUI and the event adapter.

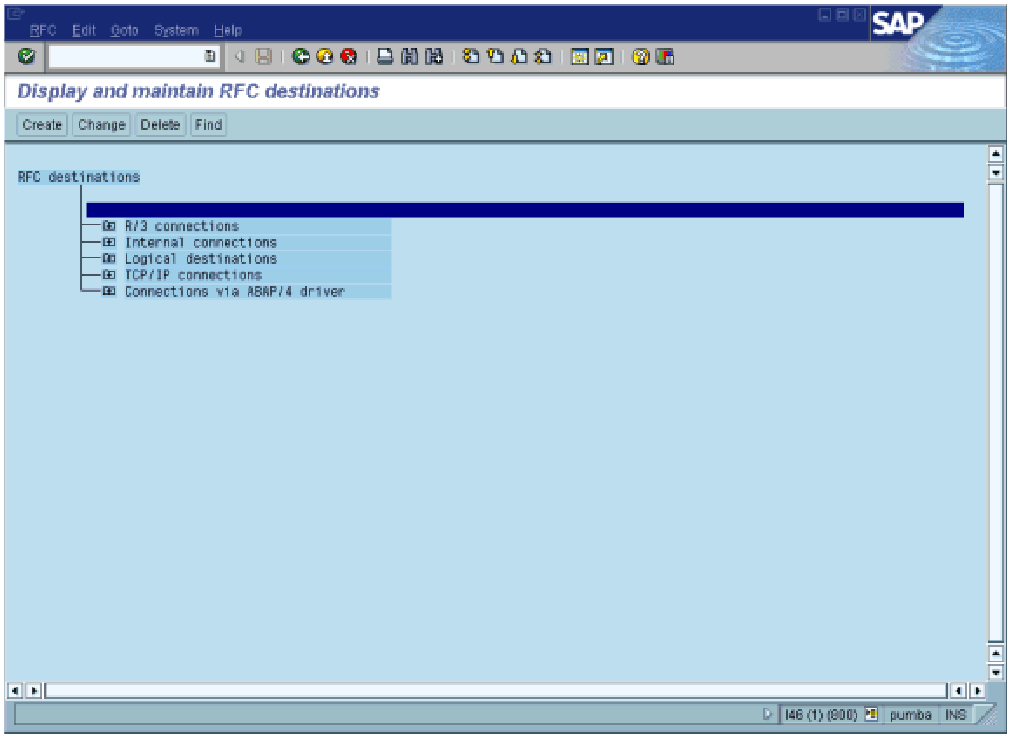
Procedure: How to Register Your Program ID

To register your program ID:

1. Launch the SAP Workbench and logon to the SAP system.
2. Select *Tools*, *Administration*, *Network*, and then *RFC destination*.

3. Execute the *SM59* transaction.

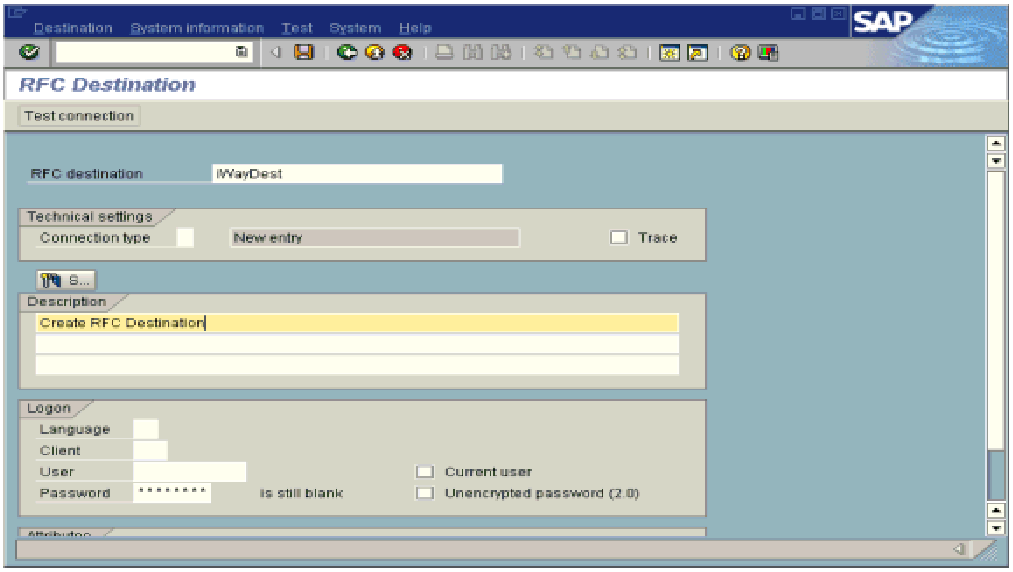
   The Display and maintain RFC destinations window opens and displays a list of connections and drivers you can manage as shown in the following image.


RFC Edit Goto System Help
Display and maintain RFC destinations
Create
Change
Delete
Find
I46 (1) (800)
pumba
INS

```
RFC destinations
    ├─ R/3 connections
    ├─ Internal connections
    ├─ Logical destinations
    ├─ TCP/IP connections
    └─ Connections via ABAP/4 driver
```

4. Select *TCP/IP connections.*
5. Click *Create.*

The RFC Destination window opens and displays fields where you provide information about the RFC destination as shown in the following image.



RFC destination: IWayDest

Connection type: New entry

Trace

Description: Create RFC Destination

Language

Client

User

Current user

Password ******** is still blank

Unencrypted password (2.0)

a. In the RFC destination field, type a name, for example, IWAYDEST.

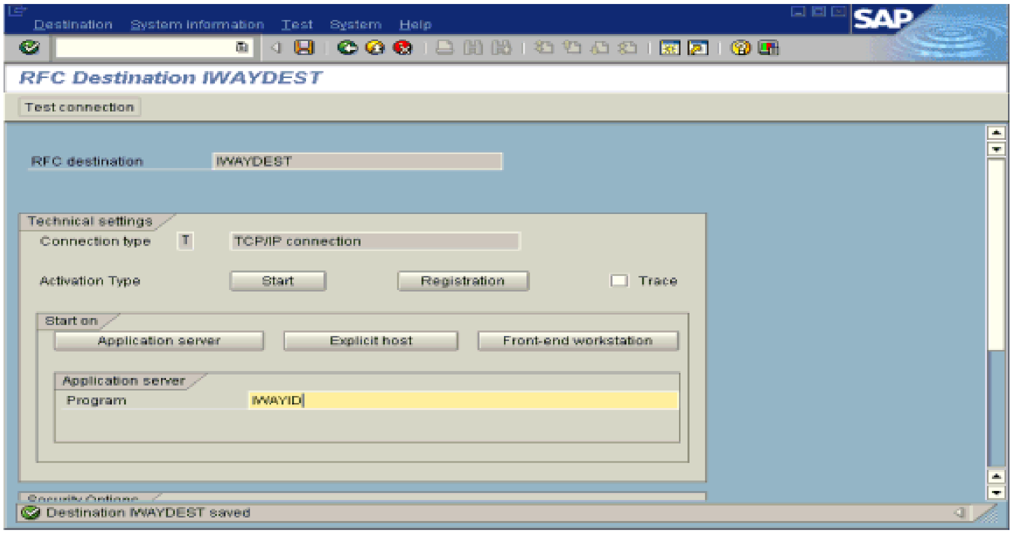
The value you type in this field is case-sensitive.

b. In the Connection type field, type *T* (for destination type, TCP/IP).

c. In the Description field, type a brief description.

6. Click *Save* from the tool bar or select *Save* from the Destination menu.

The RFC Destination IWAYDEST window opens as shown in the following image.



RFC destination IWAYDEST

Connection type T TCP/IP connection

Application server

Program IWAYID

Destination IWAYDEST saved

a. For the Activation Type, click the *Registration* button.

b. In the Program field, type *IWAYID*.

7. Click *Save* from the tool bar or select *Save* from the Destination menu.
8. Ensure your event adapter is running.
9. To verify that the SAP system and the BEA WebLogic Adapter for SAP are communicating, click *Test connection*.

Testing the SAP Event Adapter

In the SAP Server, the SE37 transaction enables you to send RFCs (Remote Function Calls) or BAPIs (Business Application Programming Interfaces) to any RFC destination. For more information on RFC destinations, see *Registering Your Program ID in SAPGUI* on page 8-3.

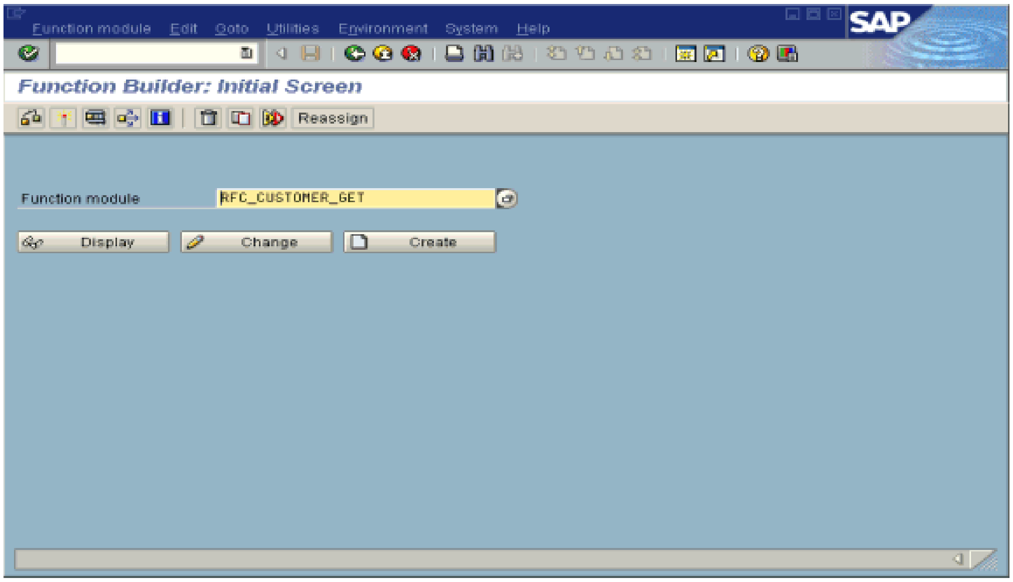
Note: Depending on the release or service pack installed, certain RFCs may not exist in your particular SAP system. Therefore, the examples included in this documentation may not be relevant to your system. If this is the case, you should use the examples as a general reference for adapter functionality and choose an RFC that exists within your SAP application environment.

Procedure: How to Test the SAP Event Adapter by Sending RFCs or BAPIs Manually

To test the SAP event adapter:

1. In the Function Builder: Initial Screen, select a function module, for example, RFC_CUSTOMER_GET.

   The following image shows the Function Builder: Initial Screen where you can select to display, change, or create a function module. RFC_CUSTOMER_GET is selected.

   
   Function module Edit Goto Utilities Environment System Help
   Function Builder: Initial Screen
   Reassign
   Display
   Change
   Create

   Function module RFC_CUSTOMER_GET

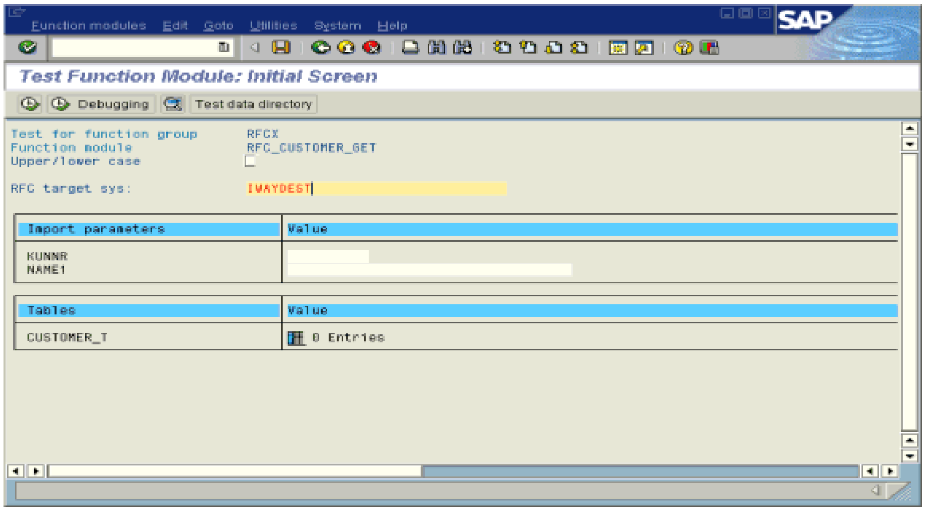
   a. To choose single test, press *F8* and click the *Single Test* icon or select *Function module*, *Test,* and then *Single Test*.

   b. Enter an RFC target system, for example, IWAYDEST.

   c. Enter input data for the particular RFC module, for example, AB*.

2. To execute, press *F8*.

The Test Function Module: Initial Screen opens as shown in the following image. It includes information about the test, the function module, and the target system. You can select the check box for Upper/lower case. The upper left pane lists the import parameters, and the upper right pane contains fields for the values. The lower left pane lists tables, and the lower right pane lists the number of entries.



Test Function Module: Initial Screen

```
Test for function group     RFCX
Function module             RFC_CUSTOMER_GET
Upper/lower case            ☐

RFC target sys:             IWAYDEST
```

Import parameters	Value
KUNNR	
NAME1	

Tables	Value
CUSTOMER_T	0 Entries

3. Enter data into the SAP GUI and click the *Execute* button.

   The function name and input data are transferred by RFC to create an XML document on theBEA WebLogic Server with the parameters input in SAPGUI.

Application Link Enabling Configuration for the Event Adapter

The SAP event adapter receives IDocs (Intermediate Documents) from SAP. To configure an SAP system to send IDocs to the SAP event adapter, you use the ALE (Application Link Enabling) configuration to:

1. Register your program ID in SAPGUI. For more information, see *Registering Your Program ID in SAPGUI* on page 8-3.
2. Define a port.

   A port identifies where to send messages. The port can be used only if an RFC destination was previously created.

   For more information on creating an RFC destination, see *Overview* on page 8-2. For more information on defining a port, see *How to Define a Port* on page 8-9.
3. Create a logical system.

One type of partner is a logical system. A logical system manages one or more RFC destinations. For more information, see *How to Create a Logical System* on page 8-10.

4. Create a partner profile.

   A partner profile is a definition of parameters for the electronic interchange of data with a trading partner using the IDoc interface. To communicate with a partner using the IDoc interface, you must create a partner profile. For more information, see *How to Create a Partner Profile* on page 8-11.

5. Create a distribution model for the partner and message type.

   You create a distribution model for the partner and message type you designated. For more information, see *How to Create a Distribution Model for the Partner and Message Type* on page 8-13.

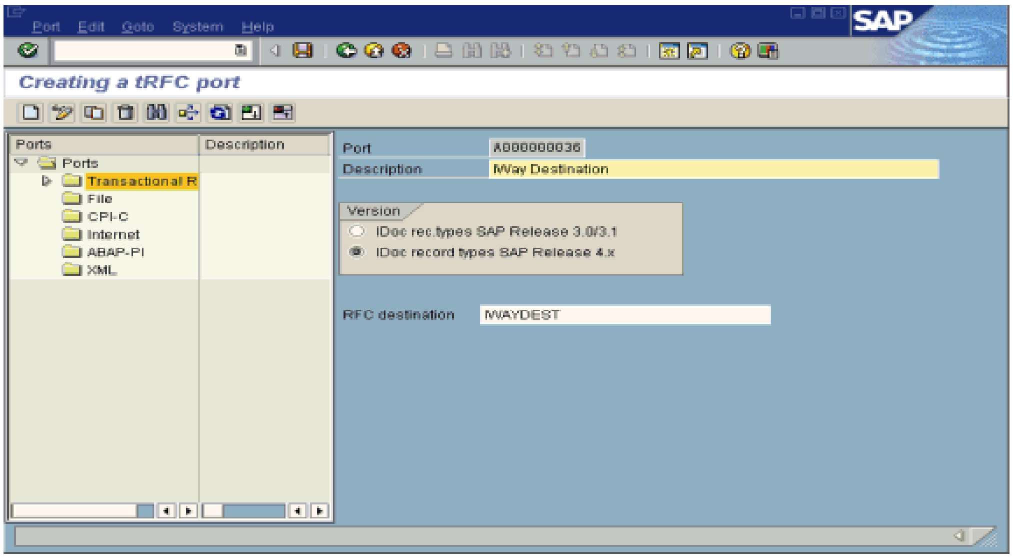
6. Test the SAP event adapter. For more information, see *Testing the SAP ALE Configuration* on page 8-16.

Procedure: How to Define a Port

To define a port:

1. In the ALE configuration, choose *Tools, Business Communications, IDocs Basis, IDoc,* and then *Port Definition* or execute the *WE21* transaction.

   The Creating a tRFC port window opens as shown in the following image. On the left, the window is divided into a Ports pane and a Description pane. A pane for displaying information about the port is on the right.



Port A000000036
Description IWay Destination
Version
( ) IDoc rec.types SAP Release 3.0/3.1
(●) IDoc record types SAP Release 4.x
RFC destination IWAYDEST

   - **a.** In the left pane under Ports, select *Transactional RFC* and click *Create*.
   - **b.** Select *Generate port name*.

     The system generates the port name.
   - **c.** In the right pane, select the IDoc version you want to send through this port.
   - **d.** Click the destination you created, for example, IWAYDEST.

2. Save the session, making note of the system-generated RFC port.

Procedure: How to Create a Logical System

To create a logical system called IWAYLOG:

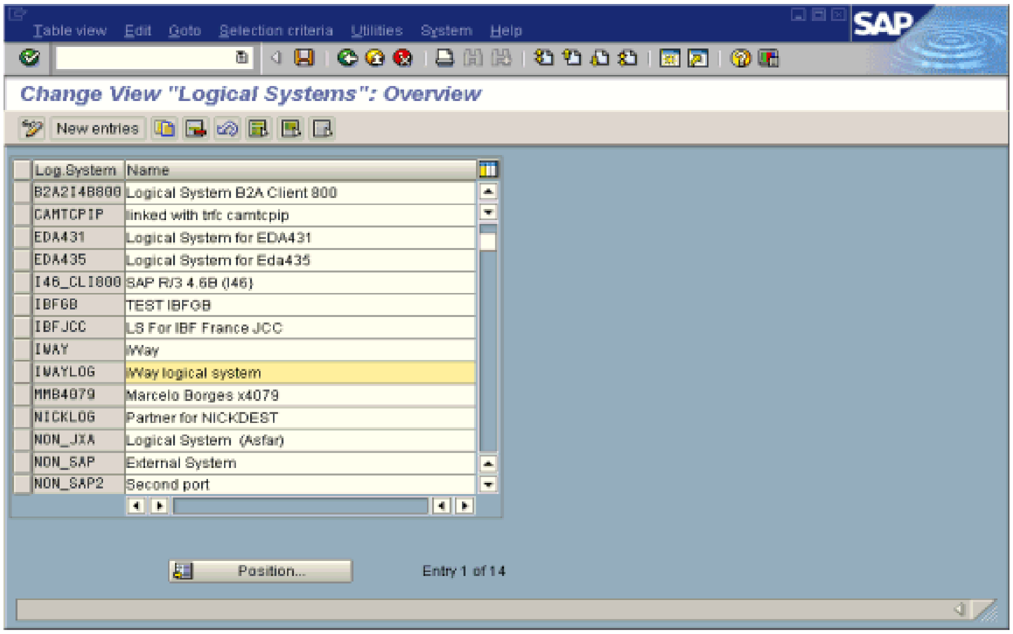
1. In the ALE Configuration, enter the area menu selection *SALE* transaction.

2. Select *SAP Reference IMG*.

3. Expand the following nodes: *Basis Components, Application Link Enabling (ALE), Sending and Receiving Systems, Logical Systems*, and *Define Logical System*.

4. Click the green check mark beside *Define Logical System*.

The Change View “Logical Systems”: Overview window opens and displays a list of logical systems and their names as shown in the following image.



Change View "Logical Systems": Overview

Log.System	Name
B2A2I4B800	Logical System B2A Client 800
CAMTCPIP	linked with trfc camtcpip
EDA431	Logical System for EDA431
EDA435	Logical System for Eda435
I46_CLI800	SAP R/3 4.6B (I46)
IBFGB	TEST IBFGB
IBFJCC	LS For IBF France JCC
IWAY	iWay
IWAYLOG	iWay logical system
MMB4079	Marcelo Borges x4079
NICKLOG	Partner for NICKDEST
NON_JXA	Logical System (Asfar)
NON_SAP	External System
NON_SAP2	Second port

Entry 1 of 14

5. Click *New entries*.

The New Entries: Overview of Added Entries window opens, as shown in the following image, with columns labelled Log.System and Name for adding new log systems.

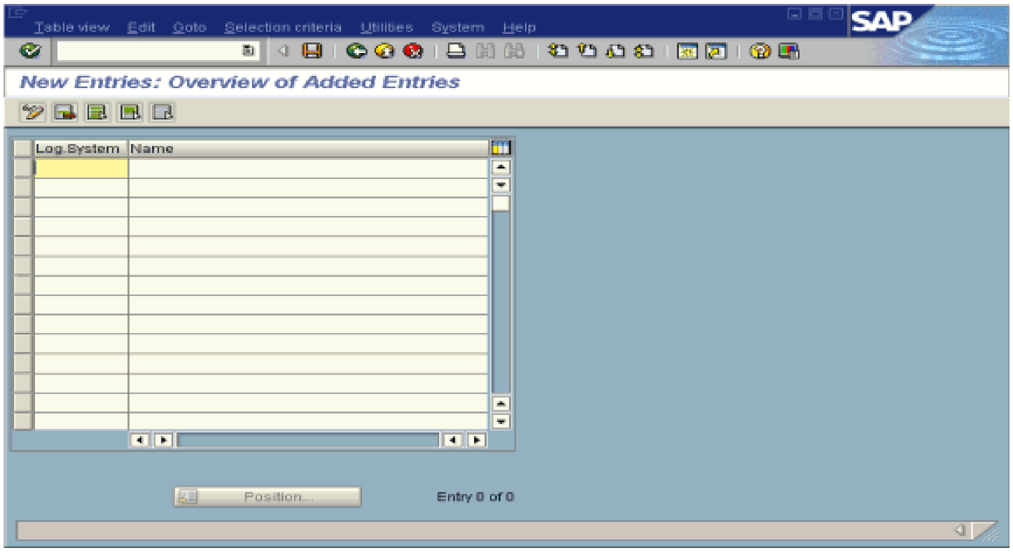


Table view Edit Goto Selection criteria Utilities System Help
New Entries: Overview of Added Entries
Log.System
Name
Position...
Entry 0 of 0

a. Type an entry for Log System, for example, IWAYLOG.

b. In the Name column, type a name (description) for the partner profile.

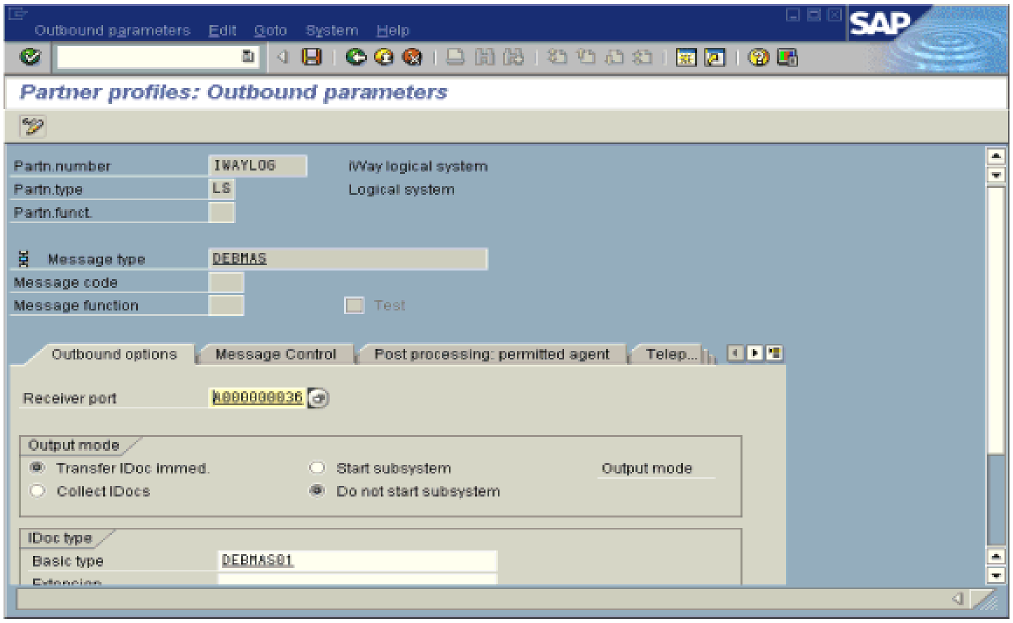
6. Save the session.

Procedure: How to Create a Partner Profile

To create a partner profile:

1. In the SAP Workbench, choose *Tools, Business Communication, IDoc Basis, IDoc,* and then *Partner profile* or execute the *WE20* transaction.

The Partner profiles: Outbound parameters window opens and displays fields for specifying details for the partner profile as shown in the following image.


Outbound parameters Edit Goto System Help
SAP
Partner profiles: Outbound parameters
Outbound options
Message Control
Post processing: permitted agent
Telep...

Partn.number IWAYLOG iWay logical system
Partn.type LS Logical system
Partn.funct.

Message type DEBMAS
Message code
Message function
Test

Receiver port A000000036

Output mode
● Transfer IDoc immed.
○ Collect IDocs
○ Start subsystem
● Do not start subsystem
Output mode

IDoc type
Basic type DEBMAS01
Extension

a. Select Partner type *LS* (Logical system).

b. Press *F5* (Create).

2. For Type, enter *USER*.
3. For Agent, enter the current user ID, or you may select another agent type.
4. Under the outbound parameter table control, select *Create outbound parameter*.

   Partn.type is LS.

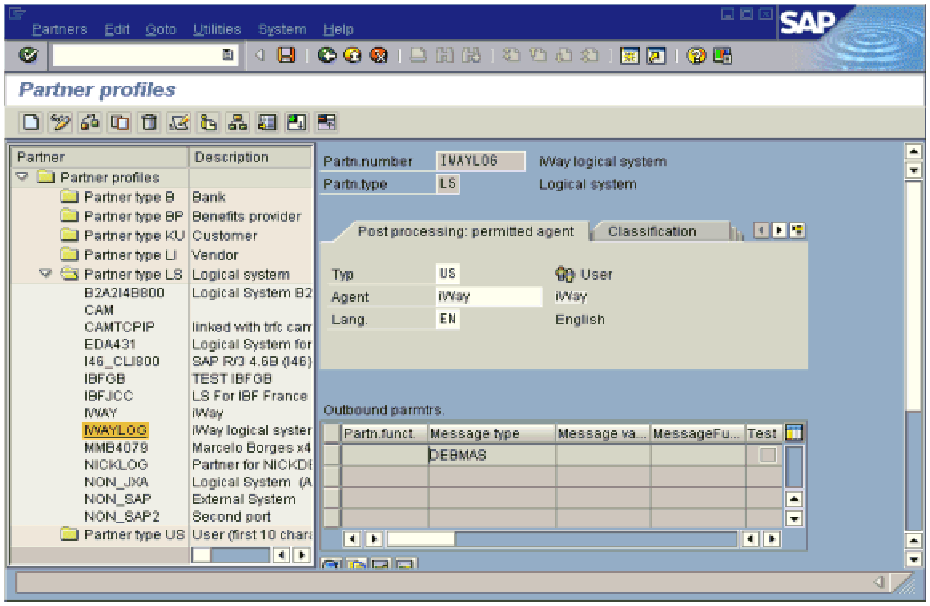
   Message type is DEBMAS (the IDoc document type).
5. Leave *Partn.funct* blank.
6. Click the *Outbound options* tab.

   a. Depending on your performance requirements, click *Transfer IDoc Immed* or *Collect IDocs*.

   b. For the IDoc, type a message type, for example, DEBMAS.

   c. Type a receiver port, for example, A000000036.
7. Save the session and exit.

The Partner profiles summary window opens and displays information for the logical system that you created as shown in the following image. In the left pane are partners and descriptions. The right pane displays information depending on which tab is active.


Partners Edit Goto Utilities System Help
SAP
Partner profiles
Post processing: permitted agent
Classification
Outbound parmtrs.

Partner	Description
Partner profiles	
Partner type B	Bank
Partner type BP	Benefits provider
Partner type KU	Customer
Partner type LI	Vendor
Partner type LS	Logical system
B2A2I4B800	Logical System B2
CAM	
CAMTCPIP	linked with trfc carr
EDA431	Logical System for
I46_CLI800	SAP R/3 4.6B (I46)
IBFGB	TEST IBFGB
IBFJCC	LS For IBF France
IWAY	iWay
IWAYLOG	iWay logical syster
MMB4079	Marcelo Borges x4
NICKLOG	Partner for NICKDI
NON_JXA	Logical System (A
NON_SAP	External System
NON_SAP2	Second port
Partner type US	User (first 10 chars

Partn.number	IWAYLOG	iWay logical system
Partn.type	LS	Logical system

Typ	US	User
Agent	iWay	iWay
Lang.	EN	English

Partn.funct.	Message type	Message va...	MessageFu...	Test
	DEBMAS			

Collected IDocs

When using collected IDocs on any platform during inbound processing (service mode), if the DOCNUM field does not have a unique document number for each IDoc, the system creates an IDoc for each header record in the collected IDoc file and duplicates the data for each IDoc.

Make sure the DOCNUM field is included in the EDI_DC40 structure and that each IDoc has a unique sequence number within the collected IDoc file.

Procedure: How to Create a Distribution Model for the Partner and Message Type

To create a distribution model called IWAYMOD:

1. In the SAP Workbench, choose *Tools, AcceleratedSAP, Customizing,* and then *Project Management* or execute the *BD64* transaction.

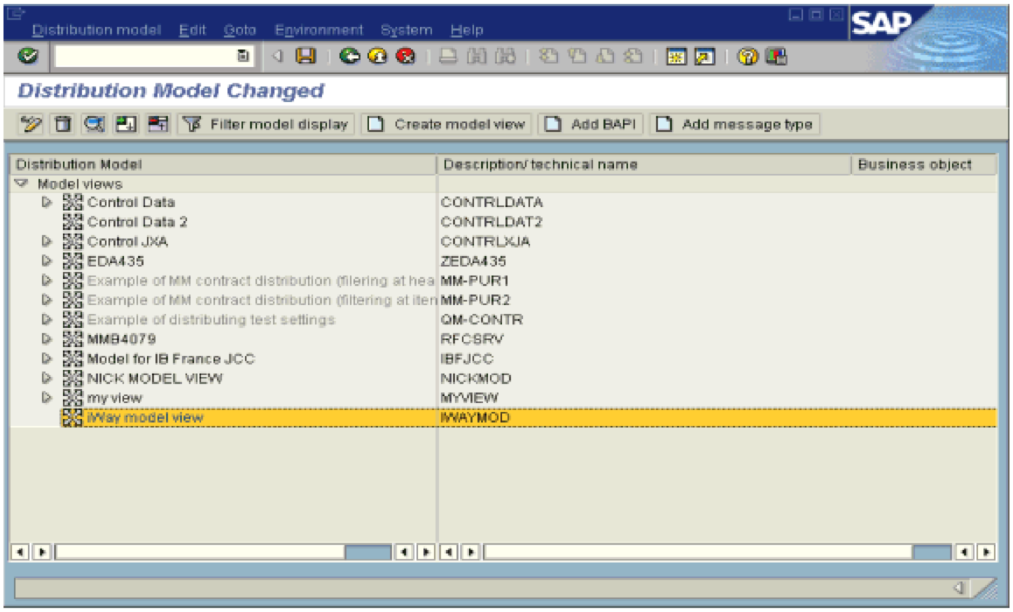
   The Display Distribution Model window opens.

2. Select *Create model view*. (If required, switch processing mode to edit within Distribution Model/Switch Processing Mode.)

3. Type a short text string and a technical name for your new model view.

4. Click the *Save* button.

   The Distribution Model Changed window opens with a tree structure of the distribution model in the left pane and the descriptions or technical names in the right pane, as shown in the following image.

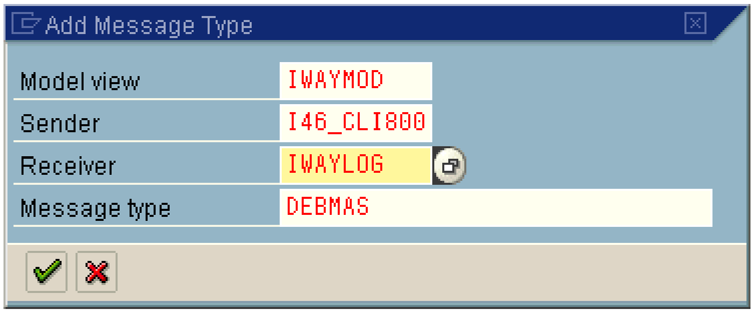


Distribution Model Changed

Distribution Model	Description/ technical name	Business object
Model views		
Control Data	CONTRLDATA	
Control Data 2	CONTRLDAT2	
Control JXA	CONTRLXJA	
EDA435	ZEDA435	
Example of MM contract distribution (filering at hea	MM-PUR1	
Example of MM contract distribution (filtering at iten	MM-PUR2	
Example of distributing test settings	QM-CONTR	
MMB4079	RFCSRV	
Model for IB France JCC	IBFJCC	
NICK MODEL VIEW	NICKMOD	
my view	MYVIEW	
iWay model view	IWAYMOD	

5. In the Distribution Model tree, select a new model view.
6. At the right, in the button bar, select *Add message type*.

   The Add Message Type pane opens and displays the name of the model view. It includes fields for specifying the sender and receiver of the message, as well as the message type, as shown in the following image.



Add Message Type

Model view	IWAYMOD
Sender	I46_CLI800
Receiver	IWAYLOG
Message type	DEBMAS

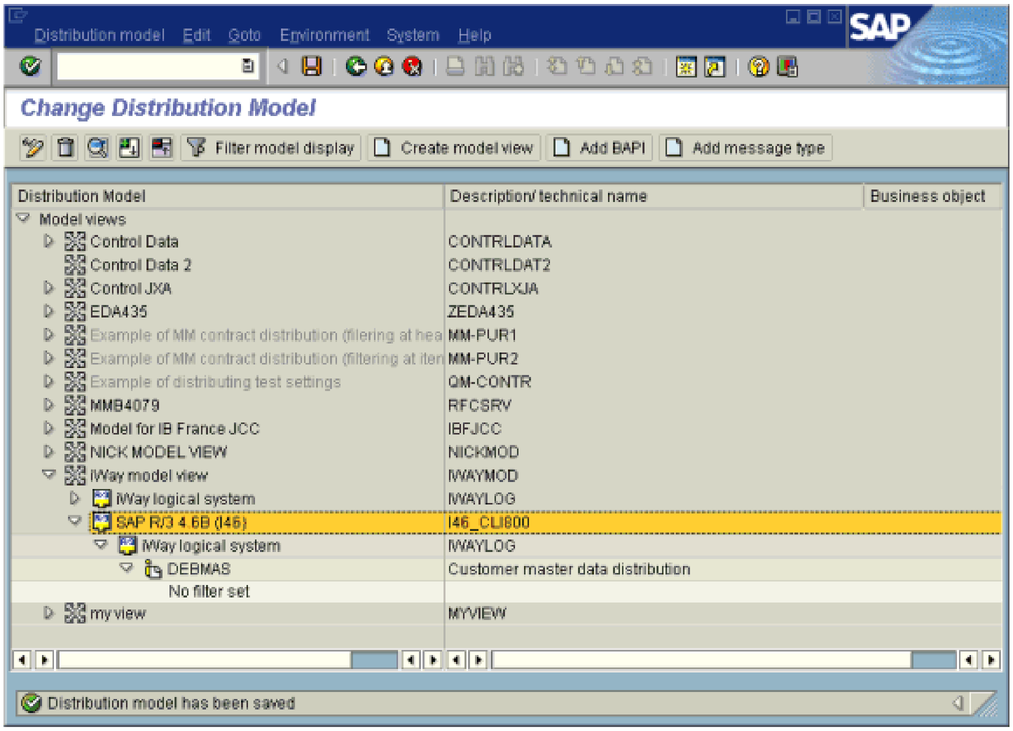
   a. In the Sender field, provide the sender that points to the SAP system that sends the IDoc, for example, I46_CLI800.

In this case, the sender is an SAP 4.6B system.

- **b.** In the Receiver field, provide the logical system, for example, IWAYLOG.
- **c.** In the Message type field, provide the type of IDoc, for example, DEBMAS.

7. Click the check mark icon.
8. Click the *Save* button.

   The Change Distribution Model window opens and displays the new model view to use to send message type, DEBMAS, from the I46_CLI800 SAP system to the IWAYLOG logical system, as shown in the following image.



Distribution Model	Description/ technical name	Business object
Model views		
Control Data	CONTRLDATA	
Control Data 2	CONTRLDAT2	
Control JXA	CONTRLXJA	
EDA435	ZEDA435	
Example of MM contract distribution (filering at hea	MM-PUR1	
Example of MM contract distribution (filtering at iten	MM-PUR2	
Example of distributing test settings	QM-CONTR	
MMB4079	RFCSRV	
Model for IB France JCC	IBFJCC	
NICK MODEL VIEW	NICKMOD	
IWay model view	IWAYMOD	
IWay logical system	IWAYLOG	
SAP R/3 4.6B (I46)	I46_CLI800	
IWay logical system	IWAYLOG	
DEBMAS	Customer master data distribution	
No filter set		
my view	MYVIEW	

Distribution model has been saved

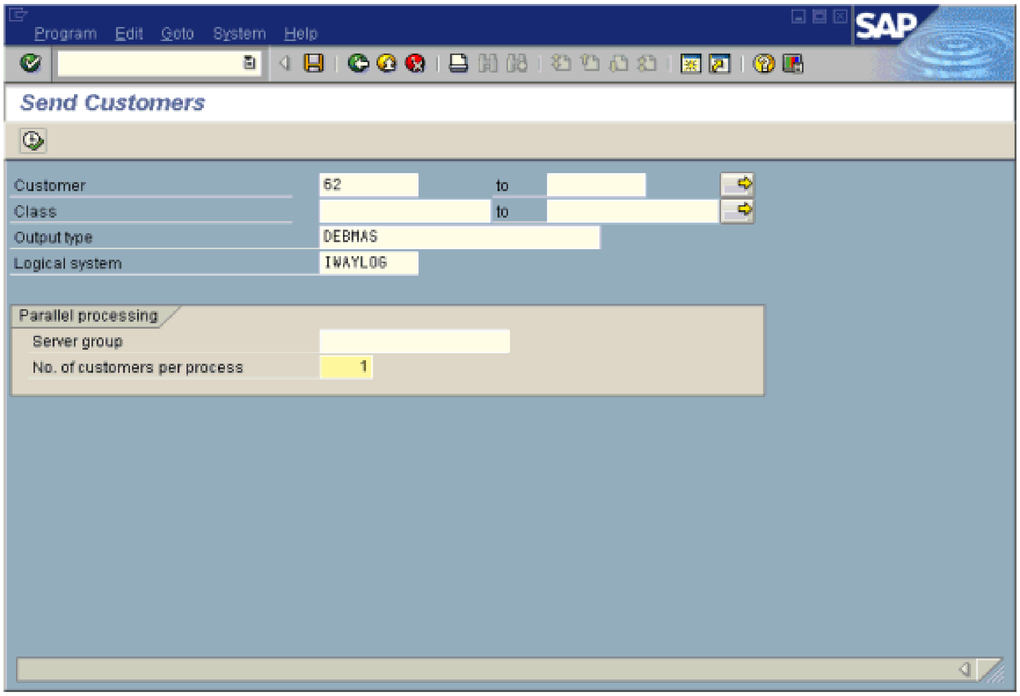
You are now ready to test the connection to the logical system.

Testing the SAP ALE Configuration

In the SAP Server, the BD12 transaction enables you to send IDocs to any logical system, for example, to an event adapter.

Procedure: How to Test the SAP ALE Configuration

The following image shows the Send Customers window where you test the message type. It includes fields for Customer, Class, Output type, and Logical system. The Parallel processing pane includes a field for Server group and a field for the number of customers per process.



Send Customers

Customer	62	to	
Class		to	
Output type	DEBMAS		
Logical system	IWAYLOG		

Parallel processing

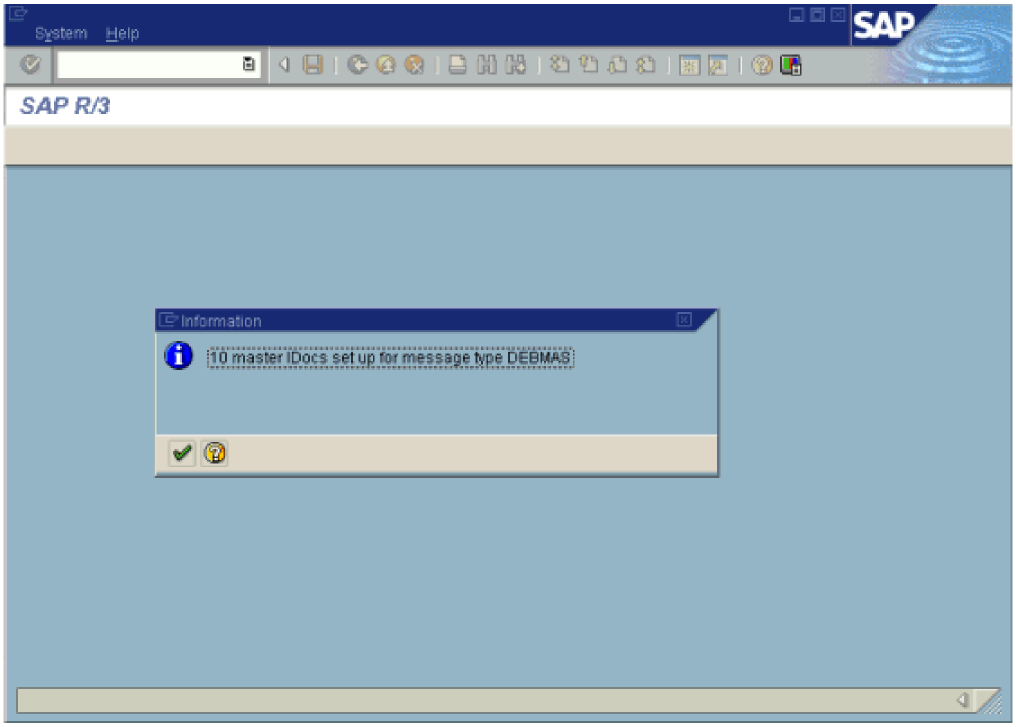
Server group	
No. of customers per process	1

To test the SAP Application Link Enabling (ALE) configuration:

1. In the Send Customers window, type the IDoc message type *DEBMAS* in the Output type field.
2. In the Logical system field, type the logical system, for example, IWAYLOG.
3. To transfer data, click the *Run* button.

   The SAP event adapter receives the IDoc in XML format. No response is expected from the event adapter.

A window opens and confirms the message entered in previous screens, as shown in the following image.



10 master IDocs set up for message type DEBMAS

Usage Considerations

The following topic provides insight for the use of SAP event handling in BEA WebLogic.

Multiple Events Using Identical Program IDs

Configuring multiple events in BEA WebLogic that use the same SAP Program ID enables you to load balance SAP event data over multiple adapter event consumers. However, when configuring each event, all the event parameters for each event must match precisely. If one of the event configuration parameters (for example, log level) is different for one of the events, BEA WebLogic creates a different schema for that event, and event data can be lost.

In this scenario, because you are using the same SAP program ID, SAP sends event data to all events configured with that program ID. BEA WebLogic creates separate schema for each event (because configuration parameter(s) differ) and then binds a specific event channel to that event schema. The result is that only events that match that schema are sent over a specific channel. While SAP is sending event data to every event configured with the same program ID, BEA WebLogic, as it validates schemas, rejects any event data that does not match the schema for that channel.

For example, only Doc. A events that appear on the first channel are received, and Doc. B events that appear on the second channel are received. Doc. B events that appear on the first channel are rejected, because they do not pass schema type validation. The same result occurs for Doc. A events that appear on the second channel.

This usage consideration applies to all supported platforms.

Workaround

If you wish to load balance SAP event data over multiple events in BEA WebLogic, ensure you configure each event in precisely the same way.

If you do not wish to use load balancing, configure a separate event for each SAP program ID. Also, ensure that no other event configured in an Application View or in another BEA application uses the same program ID.

CHAPTER 9

Troubleshooting and Error Messages

Topics:

The following topics explain limitations and workarounds when connecting to SAP. The adapter-specific errors described in this section can arise whether you are using the adapter with a JCA or with an iBSE configuration.

Troubleshooting Overview

This topic provides troubleshooting information for the BEA WebLogic Adapter for SAP, separated into four categories:

- Application Explorer
- SAP
- JCA
- iBSE

Note: The following locations include log file information that is relevant for troubleshooting.

- JCA trace information can be found under the following directory:

  `C:\Program Files\iWay55\config\base\log`
- iBSE trace information can be found under the following directory:

  `C:\Program Files\iWay55\bea\ibse\ibselogs`
- The log file for Application Explorer can be found under the following directory:

  `C:\Program File\iWay55\tools\iwae\bin`

Error Messages in Application Explorer

The following table lists errors and solutions when using Application Explorer with the adapter.

Error	**Solution**
Cannot connect to the adapter from Application Explorer.	Ensure that: SAP is running. The Server name, System Number, and Client Number are correct. The SAP user ID and password are correct.
Cannot connect to the SAP target through Application Explorer. The following error message appears: `Error getting target [SAP] - java.lang.Exception: Error Logon to SAP System`	Ensure that you enter the correct connection parameters when connecting to the SAP target.

Error	Solution
SAP does not appear in the Application Explorer adapter node list.	Ensure that you added the sapjco.jar and sapjcorfc.dll files to the lib directory. Ensure that you added the librfc32.dll file to the Windows system32 folder.
Cannot connect to your SAP system through Application Explorer. The following error message appears: `Problem activating adapter. (com.ibi.sapr3.SapAdapterException : com.sap.mw.jco.JCO$Exception: (102) RFC_ERROR_COMMUNICATION: Connect to SAP gateway failed Connect_PM GWHOST=isdsrv8, GWSERV=sapgw00, ASHOST=isdsrv8, SYSNR=00 LOCATION CPIC (TCP/IP) on local host ERROR partner not reached (host isdsrv8, service 3300) TIME Fri Aug 27 11:49:14 2004 RELEASE 620 COMPONENT NI (network interface) VERSION 36 RC -10 MODULE ninti.c LINE 979 DETAIL NiPConnect2 SYSTEM CALL SO_ERROR ERRNO 10061 ERRNO TEXT WSAECONNREFUSED: Connection refused COUNTER 1). Check logs for more information.`	Ensure that SAP is running and that the parameter values for connecting to your server are correct.

Error	**Solution**
Cannot connect to your SAP system through Application Explorer even though SAP is running. The following error message appears: `Problem activating adapter. (com.ibi.sapr3.SapAdapterException : java.lang.ExceptionInInitializerError: JCO.classInitialize(): Could not load middleware layer 'com.sap.mw.jco.rfc.MiddlewareRFC' JCO.nativeInit(): Could not initialize dynamic link library sapjcorfc [no sapjcorfc in java.library.path]. java.library.path`	Ensure that you added the sapjcorfc.dll file to the lib directory and the librfc32.dll file to the Windows system32 folder.

The DLL is loaded in another class loader (iBSE and JCA are installed on the same server). The following error message appears:
`com.ibi.sapr3.SapAdapterException: java.lang.ExceptionInInitializerError: JCO.classInitialize(): Could not load middleware layer 'com.sap.mw.jco.rfc.MiddlewareRFC'`
`JCO.nativeInit(): Could not initialize dynamic link library sapjcorfc [Native Library F:\iWay55.008.0628\lib\sapjcorfc.dll already loaded in another classloader]. java.library.path`
	Ensure that you added the sapjco.jar file to the server class path.

Error Messages in SAP

The following table lists errors and solutions when using the adapter.

Error	**Solution**
When executing a request, the following error message appears: `AdapterException: java.lang.Exception: Function module CUSTOMER_GETDETAIL2 does NOT exist.`	Check the syntax of your input XML document and ensure the name of the Remote Function module is correct and is available in SAP.
When executing a request, the following error message appears: `AdapterException: java.lang.Exception: Object type unknown for business object: CUST`	Check the syntax of your input XML document and ensure the Object type is correct.
When executing a request, the following error message appears: `AdapterException: java.lang.Exception: Unable to retrieve BAPI name for: CUSTOMER.DETAIL2`	Check the syntax of your input XML document and ensure the name of the BAPI is correct and is available in SAP.
When executing a request, the following error message appears: `java.lang.RuntimeException: com.sap.mw.jco.JCO$AbapException: (126) OBJECT_UNKNOWN: Basic type or extension does not exist.`	Check the syntax of your input XML document and ensure the IDoc extension is correct and is available in SAP.
When executing a request, the following error message appears: `AdapterException: java.lang.Exception: BapiError/BapiAbort: You are not authorized to display customers.`	Make sure your user ID has the correct permissions configured in SAP. For more information, consult your SAP administrator.

Error Messages in JCA

The following table lists an error and a solution when using JCA.

Error	Solution
In Application Explorer, the following error message appears when you attempt to connect to a JCA configuration: `Could not initialize JCA`	In the Details tab in the right pane, ensure that the directory specified in the Home field points to the correct directory, for example, `*iway_home*/lib`

Error Messages in iBSE

The following topics discuss the different types of errors that can occur when processing Web services through the Integration Business Services Engine (iBSE).

General Error Handling in iBSE

The Integration Business Services Engine (iBSE) serves as both a SOAP gateway into the adapter framework and as the engine for some of the adapters. At design time and run time, various conditions can cause errors in iBSE when Web services that use adapters are running. Some conditions and resulting errors are exposed the same way, regardless of the specific adapter; others are exposed differently, based on the adapter being used. This topic explains what to expect when you encounter the more common error conditions on an adapter-specific basis.

Usually, the SOAP gateway (*agent*) inside iBSE passes a SOAP request message to the adapter required for the Web service. If an error occurs, the way it is exposed depends on the adapter and the API or interfaces that the adapter uses. A few scenarios cause the SOAP gateway to generate a SOAP fault. In general, when the SOAP agent inside iBSE receives an invalid SOAP request, a SOAP fault element is generated in the SOAP response. The SOAP fault element contains fault string and fault code elements. The fault code contains a description of the SOAP agent error.

The following SOAP response document results when iBSE receives an invalid SOAP request:

```
<SOAP-ENV:Envelope
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
      <SOAP-ENV:Fault>
         <faultcode>SOAP-ENV:Client</faultcode>
         <faultstring>Parameter node is missing</faultstring>
      </SOAP-ENV:Fault>
   </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

In the previous example, iBSE did not receive an element in the SOAP request message that is mandatory for the WSDL for this Web service.

Adapter-Specific Error Handling

When an adapter raises an exception during run time, the SOAP agent in iBSE produces a SOAP fault element in the generated SOAP response. The SOAP fault element contains fault code and fault string elements. The fault string contains the native error description from the adapter target system. Because adapters use the target system interfaces and APIs, whether an exception is raised depends on how the target systems interface or API treats the error condition. If a SOAP request message is passed to an adapter by the SOAP agent in iBSE and that request is invalid based on the WSDL for that service, the adapter may raise an exception yielding a SOAP fault.

Although it is almost impossible to anticipate every error condition that an adapter may encounter, the following examples show how adapters handle common error conditions and how error conditions are then exposed to the Web services consumer application.

Example: BEA WebLogic Adapter for SAP Invalid SOAP Request

When the BEA WebLogic Adapter for SAP receives a SOAP request message that does not conform to the WSDL for the Web service being executed, the following SOAP response is generated:

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/
 soap/envelope/">
  <SOAP-ENV:Body>
    <SOAP-ENV:Fault>
      <faultcode>SOAP-ENV:Server</faultcode>
      <faultstring>Error processing agent [XDSapIfrAgent] - XD[FAIL]
       SapIFRException: java.sql.SQLException:
       com.ibi.sapjco.SapCallableStatement: execute() j
       java.util.NoSuchElementException</faultstring>
    </SOAP-ENV:Fault>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Example: Empty Result From SOAP Request

When the BEA WebLogic Adapter for SAP executes an SAP object as a Web service using input parameters passed in the SOAP request message that do not match records in SAP, the following SOAP response is generated:

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/
 soap/envelope/">
  <SOAP-ENV:Body>
   <SOAP-ENV:Fault>
    <faultcode>SOAP-ENV:Server</faultcode>
    <faultstring>Error processing agent [XDSapIfrAgent] - XD[FAIL]
     SapIFRException: java.sql.SQLException:
     com.ibi.sapjco.SapCallableStatement: execute()
     java.sql.SQLException: JCO Error Key: NO_RECORD_FOUND Short
     Description: com.sap.mw.jco.JCO$AbapException: (126)
     NO_RECORD_FOUND: NO_RECORD_FOUND</faultstring>
   </SOAP-ENV:Fault>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Example: Failure to Connect to SAP

When the BEA WebLogic Adapter for SAP cannot connect to SAP when executing a Web service, the following SOAP response is generated:

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/
 soap/envelope/">
  <SOAP-ENV:Body>
    <SOAP-ENV:Fault>
      <faultcode>SOAP-ENV:Server</faultcode>
      <faultstring>Error processing agent [XDSapIfrAgent] - XD[RETRY]
       Connect to SAP gateway failed Connect_PM GWHOST=ESDSUN,
       GWSERV=sapgw00, ASHOST=ESDSUN, SYSNR=00 LOCATION CPIC (TCP/IP) on
       local host ERROR partner not reached (host ESDSUN, service 3300)
       TIME Mon Jun 30 16:01:02 2003 RELEASE 620 COMPONENT NI (network
       interface) VERSION 36 RC -10 MODULE ninti.c LINE 976 DETAIL
       NiPConnect2 SYSTEM CALL SO_ERROR ERRNO 10061 ERRNO TEXT
       WSAECONNREFUSED: Connection refused COUNTER 1</faultstring>
    </SOAP-ENV:Fault>
   </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Example: Invalid SOAP Request

When the BEA WebLogic Adapter for SAP receives a SOAP request message that does not conform to the WSDL for the Web services being executed, the following SOAP response is generated:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/
 soap/envelope/">
  <SOAP-ENV:Body>
    <SOAP-ENV:Fault>
      <faultcode>SOAP-ENV:Server</faultcode>
      <faultstring>RPC server connection failed: Connection refused:
       connect
      </faultstring>
    </SOAP-ENV:Fault>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Example: Empty Result From an BEA WebLogic Adapter for SAP SOAP Request

Note: The condition for this adapter does not yield a SOAP fault.

When the BEA WebLogic Adapter for SAP executes a SOAP request using input parameters passed that do not match records in the target system, the following SOAP response is generated:

```
<SOAP-ENV:Envelope xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance"
 xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
 xmlns:xsd="http://www.w3.org/1999/XMLSchema">
   <SOAP-ENV:Body>
      <m:RunDBQueryResponse xmlns:m="urn:schemas-iwaysoftware-com:iwse"
        xmlns="urn:schemas-iwaysoftware-com:iwse"
        cid="2A3CB42703EB20203F91951B89F3C5AF">
        <RunDBQueryResult run="1" />
     </m:RunDBQueryResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

APPENDIX A

Using Application Explorer in BEA WebLogic Workshop to Create XML Schemas and Web Services

Topics:

- Starting Application Explorer in BEA WebLogic WorkShop
- Creating a New Configuration for iBSE or JCA
- Connecting to SAP
- Viewing an Application System Object and Creating an XML Schema
- Creating an Integration Business Service
- Adding a Control for a Resource in BEA WebLogic Workshop
- Extensible CCI Control

This section describes how to use the Java Swing Application Explorer in BEA WebLogic Workshop to create XML schemas for SAP BAPIs, RFCs, and IDocs. In addition, it provides information on creating Web services that are published by the Integration Business Services Engine (iBSE).

Although this section describes the Java Swing implementation of Application Explorer, other implementations provide the same functionality using similar graphical user interfaces. For more information, see the following chapters:

- Chapter 3, *Creating XML Schemas for SAP*
- Chapter 4, *Creating and Publishing Integration Business Services*

Starting Application Explorer in BEA WebLogic WorkShop

The server must be started where Application Explorer is running. Before you can use Application Explorer, you must start BEA WebLogic server.

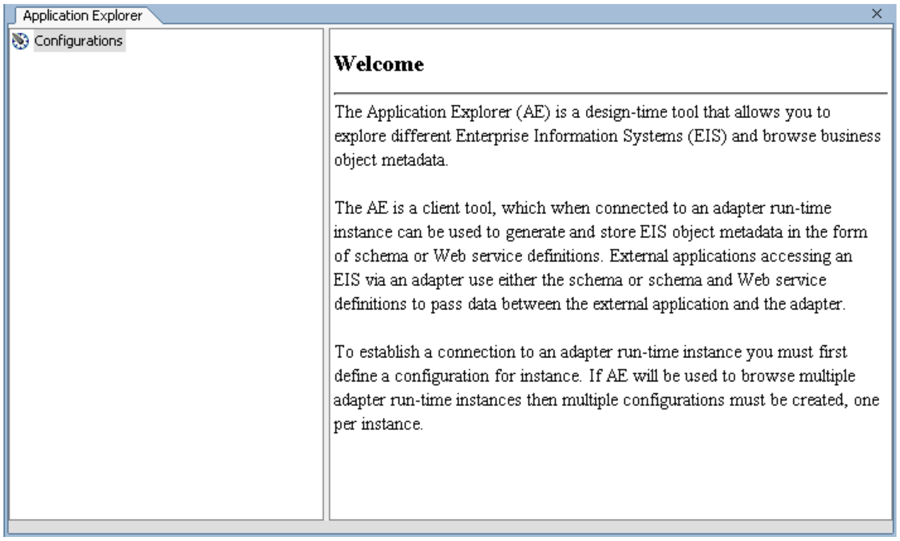
You can run Application Explorer in BEA WebLogic Workshop using an Integration Business Services Engine (iBSE) configuration. If you want to use Application Explorer with a JCA configuration instead of iBSE, you must use the servlet version of Application Explorer that runs outside of WebLogic Workshop. For more information about the servlet version, see Chapter 3, *Creating XML Schemas for SAP*.

Procedure How to Start Application Explorer

To start Application Explorer running in BEA WebLogic Workshop:

1. Ensure the server on which Application Explorer is deployed is started.
2. Start BEA WebLogic Workshop.
3. From the BEA WebLogic Workshop View menu, select *Windows* and then, *Application Explorer.*

   Application Explorer opens in BEA WebLogic Workshop.



Welcome

The Application Explorer (AE) is a design-time tool that allows you to explore different Enterprise Information Systems (EIS) and browse business object metadata.

The AE is a client tool, which when connected to an adapter run-time instance can be used to generate and store EIS object metadata in the form of schema or Web service definitions. External applications accessing an EIS via an adapter use either the schema or schema and Web service definitions to pass data between the external application and the adapter.

To establish a connection to an adapter run-time instance you must first define a configuration for instance. If AE will be used to browse multiple adapter run-time instances then multiple configurations must be created, one per instance.

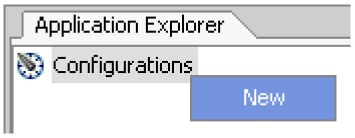
You can resize and drag-and-drop the Application Explorer window within BEA WebLogic Workshop. For example, you can drag it to the upper part of BEA WebLogic Workshop.

Creating a New Configuration for iBSE or JCA

Before you can start using Application Explorer, you must define a new configuration for iBSE or JCA.

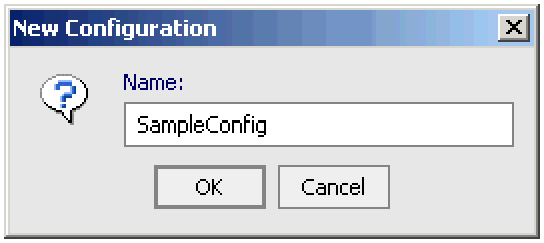
Procedure How to Create a New Configuration for iBSE or JCA

To create a new configuration:



1. Right-click *Configurations* and select *New*.

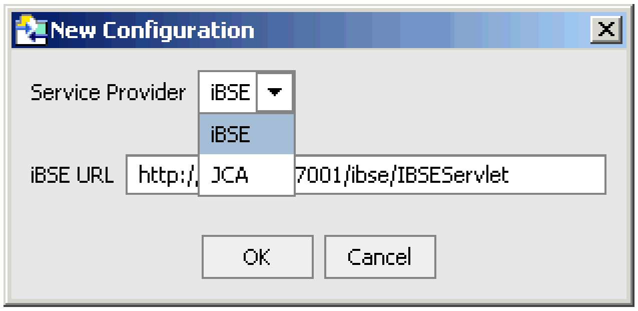
   The New Configuration dialog box opens.



2. Type the name of the new configuration and click *OK*.

   Note: If you are creating a new JCA configuration, type *base* in the name field. You must use this value if you are pointing to the default configuration.

   The following dialog box opens.



3. From the Service Provider drop-down list, select *iBSE* or *JCA*.

   - If you select iBSE, type the URL for iBSE, for example, `http://localhost:7001/ibse/IBSEServlet`

where:

`localhost`

Is where your application server is running.

- If you select JCA, enter the full path to the directory where iWay 5.5 is installed, for example,

`C:\Program Files\iWay55`

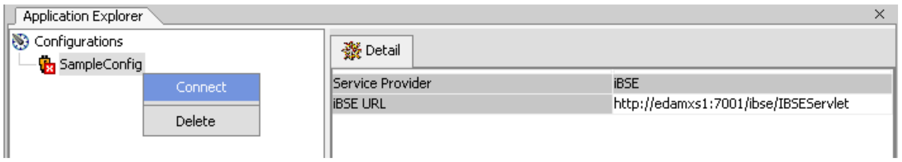
where:

`iWay55`

Is the full path to your iWay installation.

A node representing the new configuration appears under the Configurations node. The right pane provides details of the configuration you created.

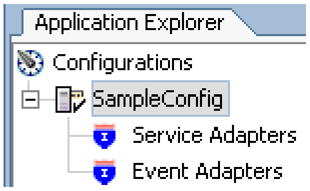
After you add your configuration, you must connect to it.



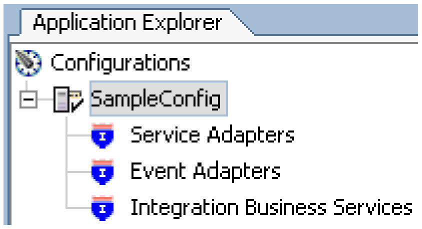
Service Provider	iBSE
iBSE URL	http://edamxs1:7001/ibse/IBSEServlet

4. Right-click the configuration to which you want to connect, for example, base, and select *Connect*.

   The Service Adapters and Event Adapters nodes appear.



When you connect to iBSE, the Service Adapters, Event Adapters, and Integration Business Services nodes appear.



5. To display the service and event adapters that are installed, expand each node.

   The Service Adapters list includes an SAP node that enables you to connect to SAP metadata and create XML request and response schemas to use to listen for events or create Web services. For more information, see *Creating an Integration Business Service* on page A-17.

   The Event Adapters list includes an SAP node that enables you to create ports and channels for SAP event handling. For more information, see Appendix B, *Using Application Explorer in BEA WebLogic Workshop for Event Handling*.

Connecting to SAP

To browse BAPIs, RFCs, and IDocs in SAP, you must create a target for SAP. The target serves as your connection point and is automatically saved after it is created. You must establish a connection to SAP every time you start Application Explorer or after you disconnect from SAP.

The left pane displays the application systems supported by Application Explorer based on the adapters you installed and are licensed to use.

Creating and Connecting to a Target

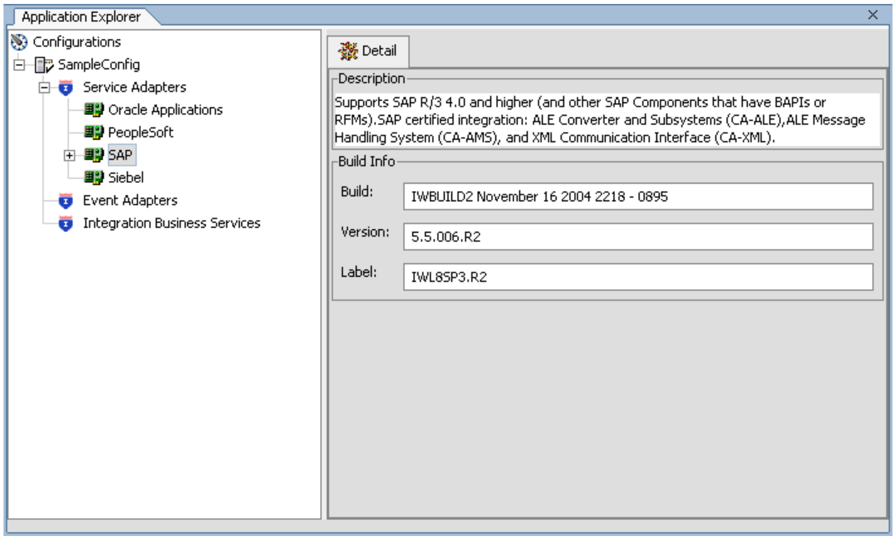
To connect to SAP, you must create and connect to a target using Application Explorer.

Procedure How to Create a New Target for SAP

To create a target for SAP:

1. In the left pane, expand *Service Adapters* and click the *SAP* node.

Descriptive information (for example, title and product version) for the BEA WebLogic Adapter for SAP appears in the right pane.



Description

Supports SAP R/3 4.0 and higher (and other SAP Components that have BAPIs or RFMs).SAP certified integration: ALE Converter and Subsystems (CA-ALE),ALE Message Handling System (CA-AMS), and XML Communication Interface (CA-XML).

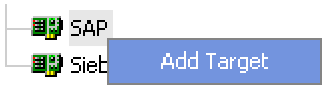
Build Info

Build: IWBUILD2 November 16 2004 2218 - 0895

Version: 5.5.006.R2

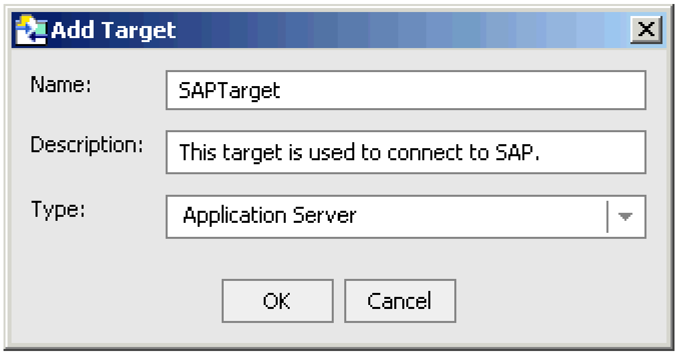
Label: IWL8SP3.R2

2. To view the options, right-click the *SAP* adapter node.



3. Select *Add Target*.

   The Add target dialog box opens.



Name: SAPTarget

Description: This target is used to connect to SAP.

Type: Application Server

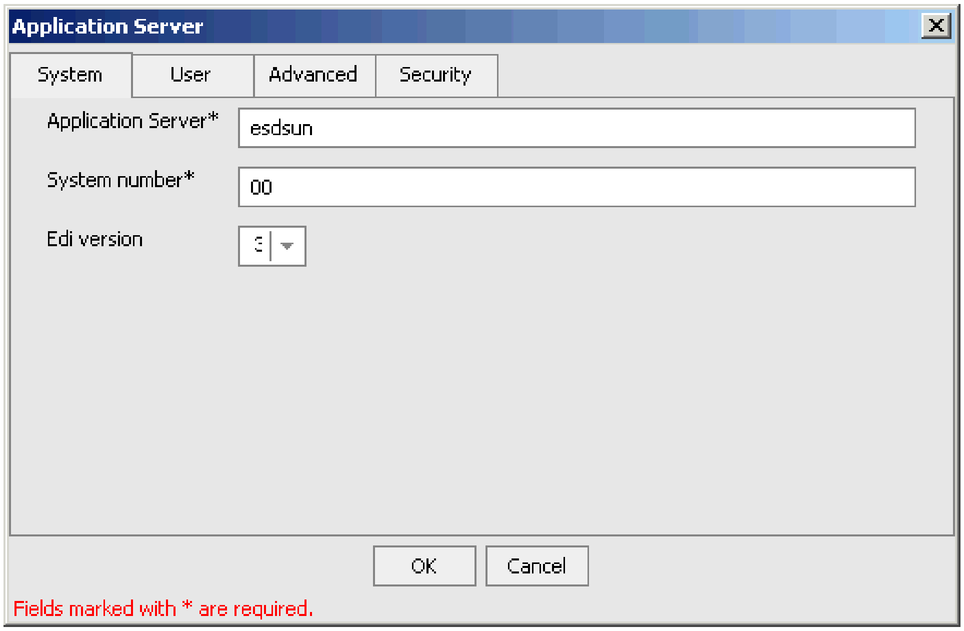
   a. In the Name field, type a descriptive name for the target, for example, SAPTarget.
   b. In the Description field, enter a brief description for the target.

   c. From the Type drop-down list, select the type of server to which you are connecting.

      Application Server is the default value.

4. Click *OK*.

   The Application Server dialog box opens where you must specify connection information for SAP and the application server that is hosting SAP.



Application Server* esdsun

System number* 00

Edi version 3

Fields marked with * are required.

5. Enter the following SAP system information on the System tab.

   a. In the Application Server field, type the host name or IP address for the machine that is hosting the SAP application (required).

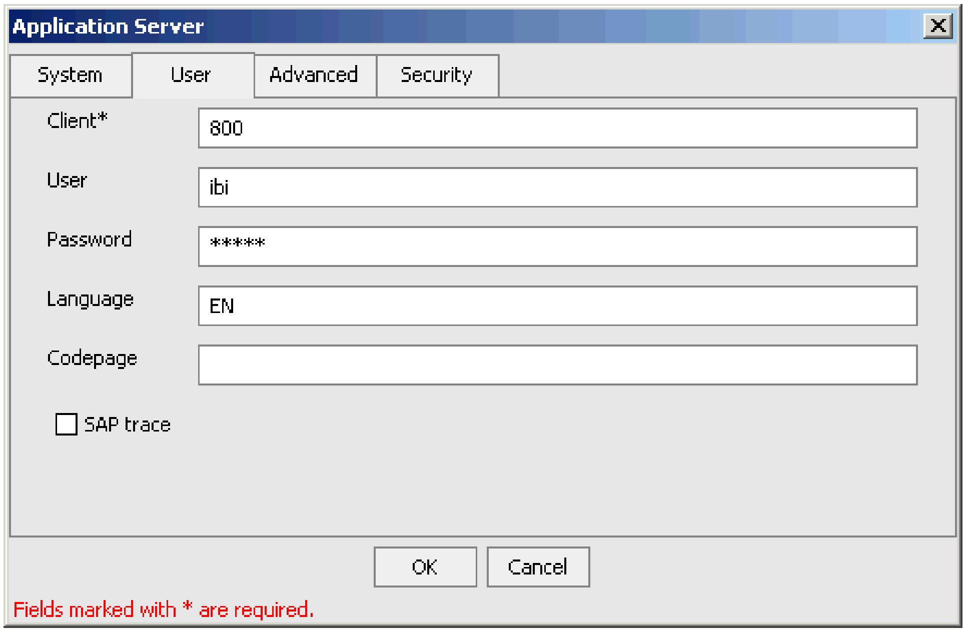
   b. In the System number field, type the system number defined to SAP for client communications (required).

   c. From the EDI version drop-down list, select the Electronic Data Interchange (EDI) document version that you are using with the BEA WebLogic Adapter for SAP.

      Version 3 is the default value.

6. To view and provide information on the User tab, click *User*.

The User tab becomes active.



Client* 800

User ibi

Password *****

Language EN

Codepage

☐ SAP trace

Fields marked with * are required.

a. In the Client field, type the client number defined for the SAP application for client communications (required).

b. In the User field, type a valid user ID for the SAP application.

c. In the Password field, type a valid password for the SAP application.

d. In the Language field, type a language key.

EN (English) is the default value.

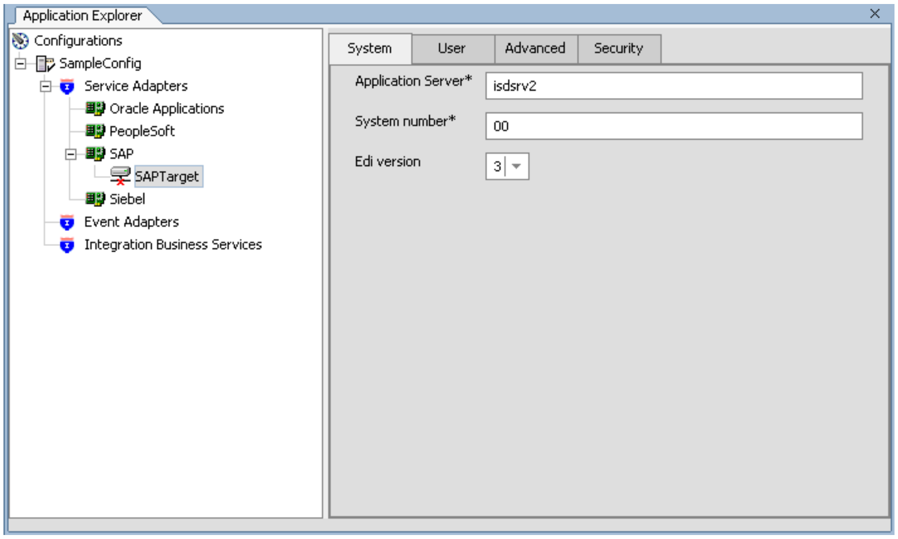
e. In the Codepage field, type a character code page value.

f. To enable traces, select the *SAP trace* check box.

After you provide information for the System and User tabs, you have completed the SAP target configuration. However, you can specify additional parameters in the Advanced and Security tabs.

7. Click *OK*.

The new target (SAPTarget) appears in the left pane under the SAP node.



Application Server* isdsrv2

System number* 00

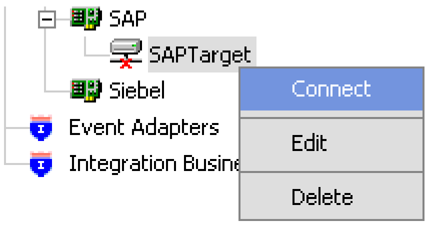
Edi version 3

In the right pane, you can review the connection information you specified. You are ready to connect to the application target you defined.

Procedure How to Connect to a Target

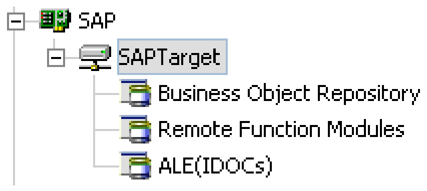
To connect to an Enterprise Information System (EIS), for example, SAP:

1. In the left pane, expand the *SAP* node and select the target to which you want to connect, for example, SAPTarget.
2. In the right pane, click the *User* tab and type a valid password for the SAP application.



3. In the left pane, right-click the target and select *Connect*.

   The SAPTarget node in the left pane changes to reflect that a connection was made.



4. Expand the target node to reveal the list of SAP business objects.

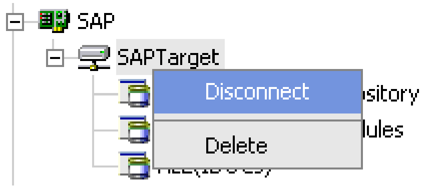
Managing a Target

Although you can maintain multiple open connections to different application systems, BEA Systems, Inc. recommends that you close connections when they are not in use. After you disconnect, you can modify an existing target.

You can modify the connection parameters when your system properties change. You also can delete a target. The following procedures describe how to disconnect from a target, edit a target, and delete a target.

Procedure How to Disconnect From a Target

To disconnect from a target:



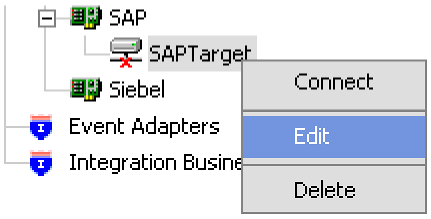
1. Right-click the target from which you want to disconnect.
2. Select *Disconnect*.

   Disconnecting from the application system drops the connection, but the node remains. The SAPTarget node in the left pane changes to reflect that you disconnected from the target.

Procedure How to Edit a Target

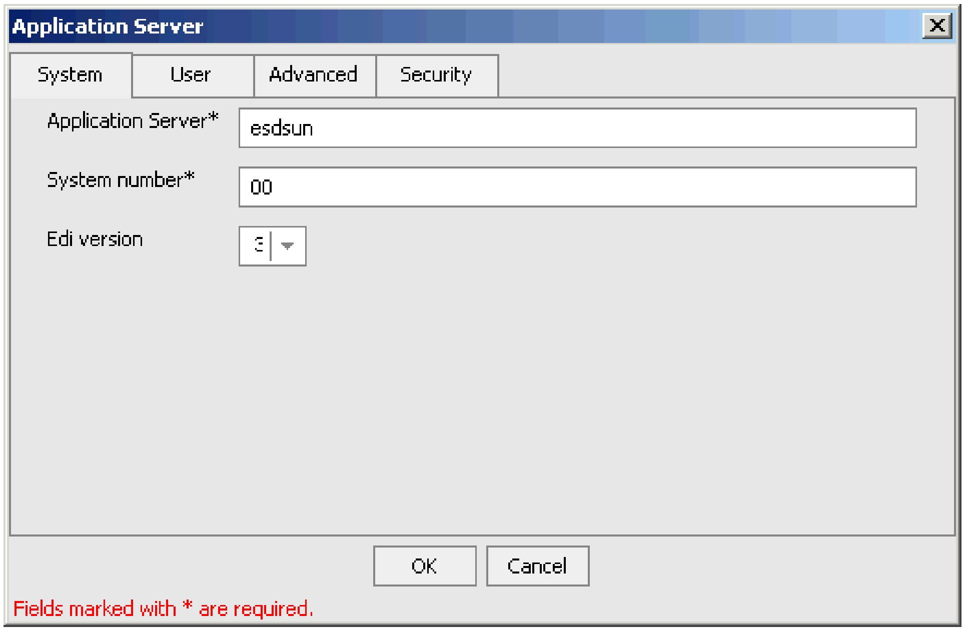
To edit a target:

1. Ensure that the target you want to edit is disconnected.



2. In the left pane, right-click the target and select *Edit*.

   The Application Server dialog box opens.



Application Server* esdsun

System number* 00

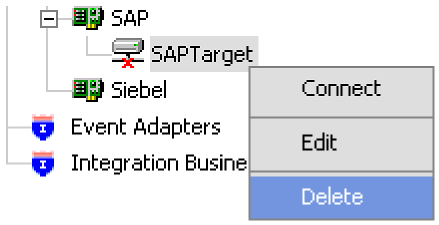
Edi version 3

Fields marked with * are required.

3. Change the properties in the dialog box as required and click *OK*.

Procedure **How to Delete a Target**

To delete a target:



1. In the left pane, right-click the target, for example, SAPTarget.
2. Select *Delete*.

   The SAPTarget node disappears from the left pane.

Viewing an Application System Object and Creating an XML Schema

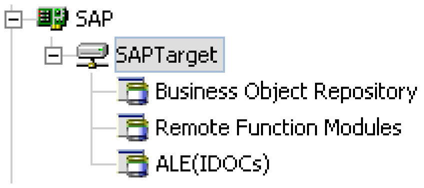
After you create a new configuration and connect to SAP, Application Explorer enables you to explore and browse business object metadata. For example, Application Explorer enables you to view BAPI, RFC, and iDOC metadata stored in the SAP Business Object repository.

Procedure **How to View an SAP System Object**

To view an application system object:

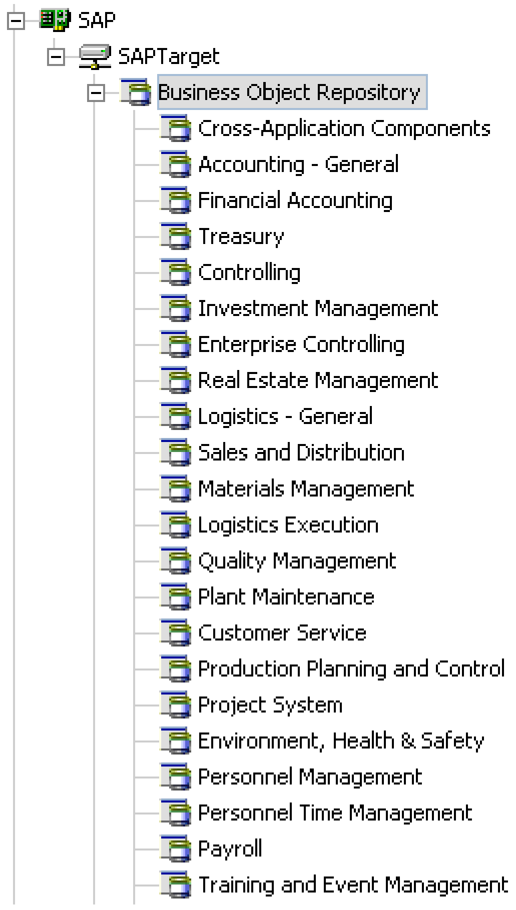
1. Click the icon to the left of the target name, for example, SAPTarget.

   The available system objects appear.



2. To expand the SAP repository node you want to explore, click the icon to the left of the repository name, for example, Business Object Repository.

A list of business object groups appears.



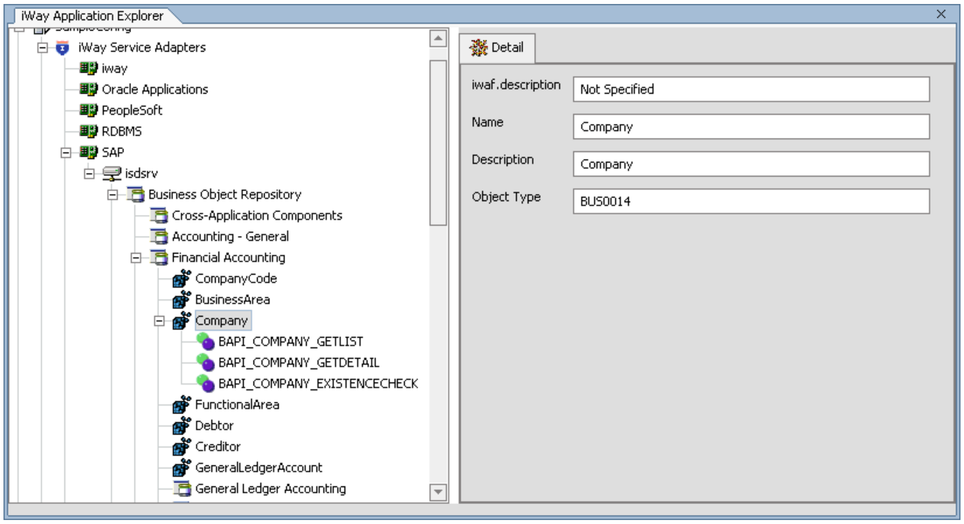
SAP
SAPTarget
Business Object Repository
Cross-Application Components
Accounting - General
Financial Accounting
Treasury
Controlling
Investment Management
Enterprise Controlling
Real Estate Management
Logistics - General
Sales and Distribution
Materials Management
Logistics Execution
Quality Management
Plant Maintenance
Customer Service
Production Planning and Control
Project System
Environment, Health & Safety
Personnel Management
Personnel Time Management
Payroll
Training and Event Management

3. Expand the *Financial Accounting* group.

   A list of business objects related to Financial Accounting appears.

4. Scroll down and click the *Company* business object.

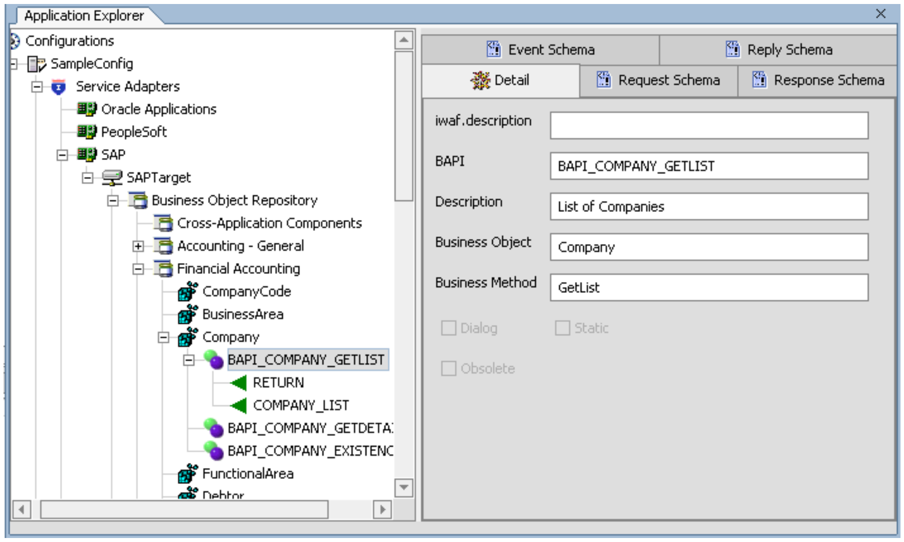
In the left pane, the following list of BAPI methods related to Company appears:



iwaf.description	Not Specified
Name	Company
Description	Company
Object Type	BUS0014

5. Click the BAPI method named *BAPI_COMPANY_GETLIST.*

In the right pane, properties for the BAPI_COMPANY_GETLIST method appear in the Detail tab.



iwaf.description	
BAPI	BAPI_COMPANY_GETLIST
Description	List of Companies
Business Object	Company
Business Method	GetList

☐ Dialog ☐ Static

☐ Obsolete

To view the following XML schemas for the method, click the corresponding tab in the right pane.

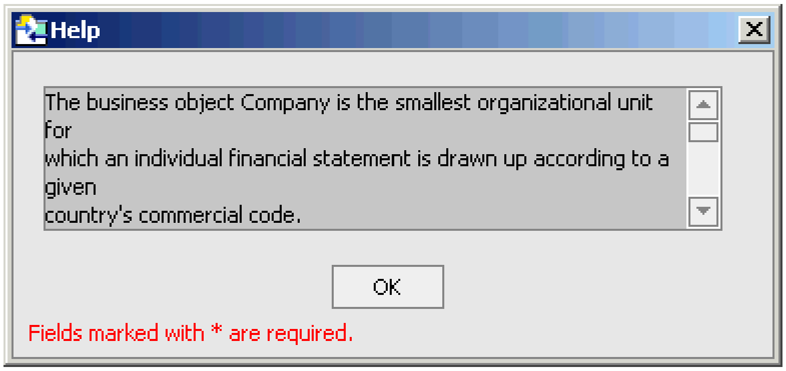
- Request
- Response
- Reply
- Event

Procedure How to View Additional Information for a Group or System Object in SAP

To view additional information for a particular group or object in SAP:

1. Right-click a component, for example, Company, and select *Help*.

   The Help window opens.

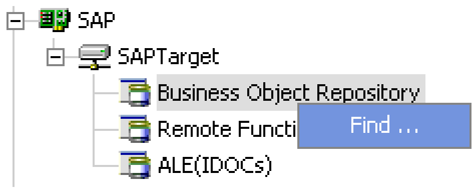


The business object Company is the smallest organizational unit for
which an individual financial statement is drawn up according to a given
country's commercial code.

2. After you finish viewing the information, click *OK*.

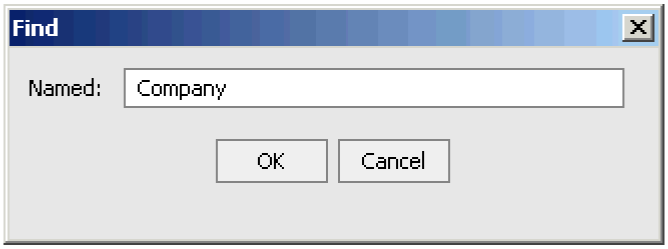
Procedure How to Search for a System Object in SAP

To narrow your search for a system object in SAP:



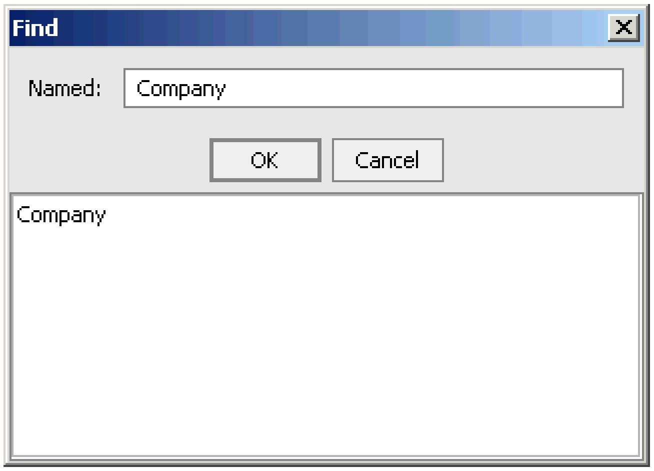
1. Right-click the system object category, for example, Business Object Repository and click *Find*.

The Find dialog box opens.



2. Type the group name you are searching for and click *OK*.

   If a match is found, the Find dialog box expands to display the results.



   If no matches are found, a message appears that indicates that no matches exist.

3. To automatically expand the group and view all the methods that are available in the left pane of Application Explorer, click the group name in the list of results.

Reference Schema Location

After you browse the application system business object repository and select a specific method, the relevant XML schemas automatically are created for that method and stored in the repository you created, for example:

```
drive:\Program Files\iWay55\bea\ibse\wsdl\schemas\service\SAP
\SAPTarget\S5710F9F
```

where

`SAPTarget`

Is the name of the SAP target.

`S5710F9F`

Is a randomly generated folder name where the schemas are stored.

Creating an Integration Business Service

Java Swing Application Explorer provides Web developers with a simple, consistent mechanism for extending the capabilities of the adapter. The Integration Business Services Engine (iBSE) exposes functionality as Web services. It serves as a gateway to heterogeneous back-end applications and databases.

A Web service is a self-contained, modularized function that can be published and accessed across a network using open standards. It is the implementation of an interface by a component and is an executable entity. For the caller or sender, a Web service can be considered as a "black box" that may require input and delivers a result. Web services integrate within an enterprise as well as across enterprises on any communication technology stack, whether asynchronous or synchronous, in any format.

After you browse the application system business object repository and create an XML schema for the object, you can generate a business service for the object you wish to use with your adapter.

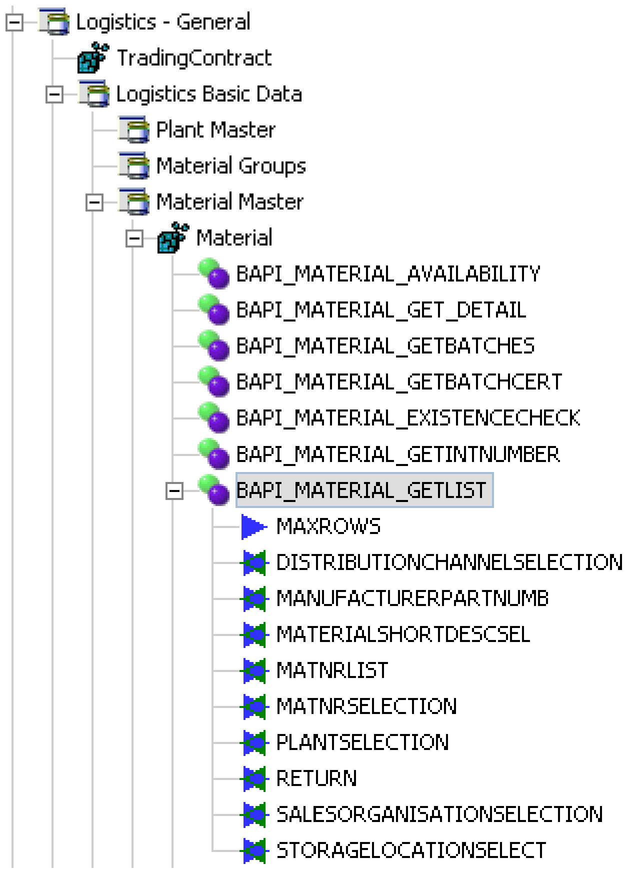
You can generate a business service (also known as a Web service) for objects you wish to use with your adapter. After creating the Web service, you can export the WSDL file for the Web service to a directory accessible by BEA WebLogic Workshop, making it easy to incorporate Web services into BEA WebLogic Workshop workflows.

Note: In a J2EE Connector Architecture (JCA) implementation of adapters, Web services are not available. When the adapters are deployed to use the Connector for JCA, the Common Client Interface provides integration services using the adapters. For more information, see *BEA WebLogic ERP Adapter Installation and Configuration*.

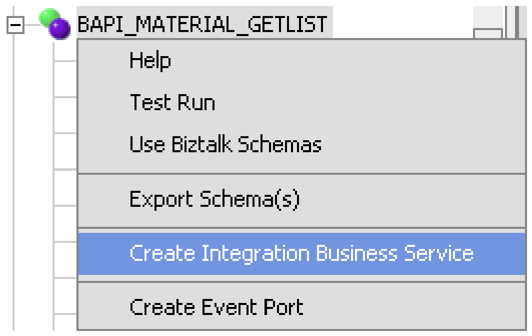
The following procedures use the SAP BAPI method called BAPI_MATERIAL_GETLIST as an example that returns a list of materials from SAP.

Procedure How to Create an Integration Business Service

To create an Integration Business Service:

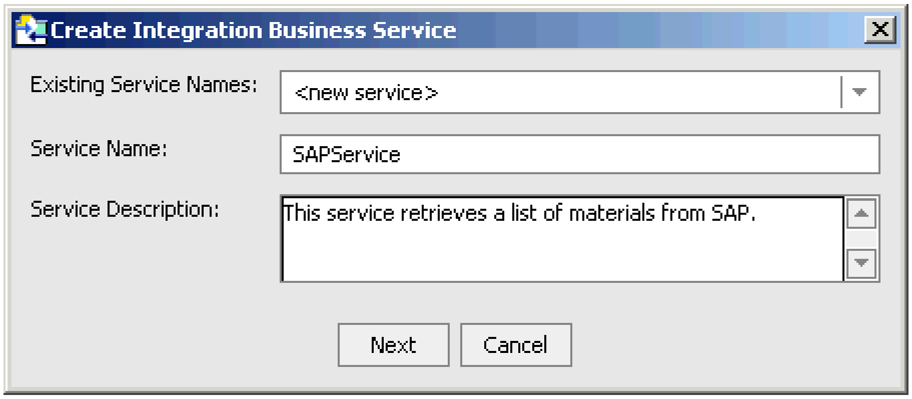


1. Select the *BAPI_MATERIAL_GETLIST* method from the Business Object Repository.



2. Right-click the method and select *Create Integration Business Service*.

   The Create Integration Business Service dialog box opens.



Existing Service Names: <new service>

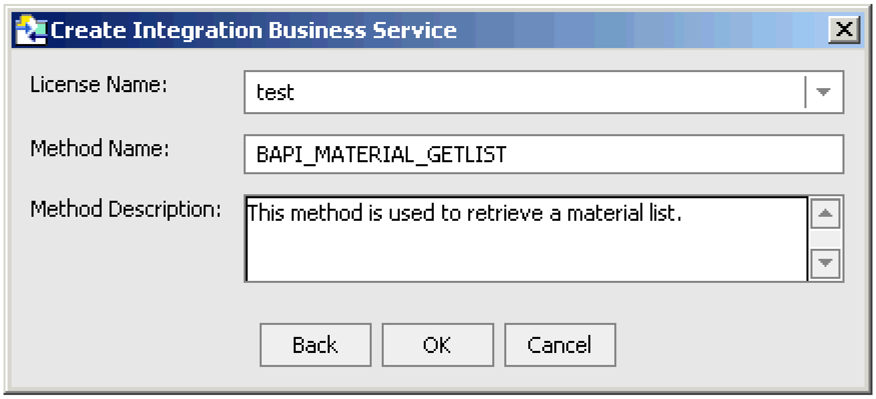
Service Name: SAPService

Service Description: This service retrieves a list of materials from SAP.

   a. From the Existing Service Names drop-down list, select whether you want to create a new service name or use an existing service name.
   b. In the Service Name field, type a name for the business service, for example, SAPService.
   c. In the Service Description field, type a brief description for the business service.

3. Click *Next*.

   The Create Integration Business Service dialog box displays additional fields.



License Name: test

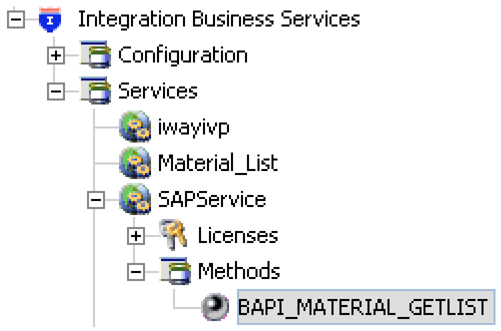
Method Name: BAPI_MATERIAL_GETLIST

Method Description: This method is used to retrieve a material list.

   a. From the License Name drop-down list, select a license.
   b. In the Method Name field, type a name for the method, for example, BAPI_MATERIAL_GETLIST.
   c. In the Method Description field, type a brief description for the method.

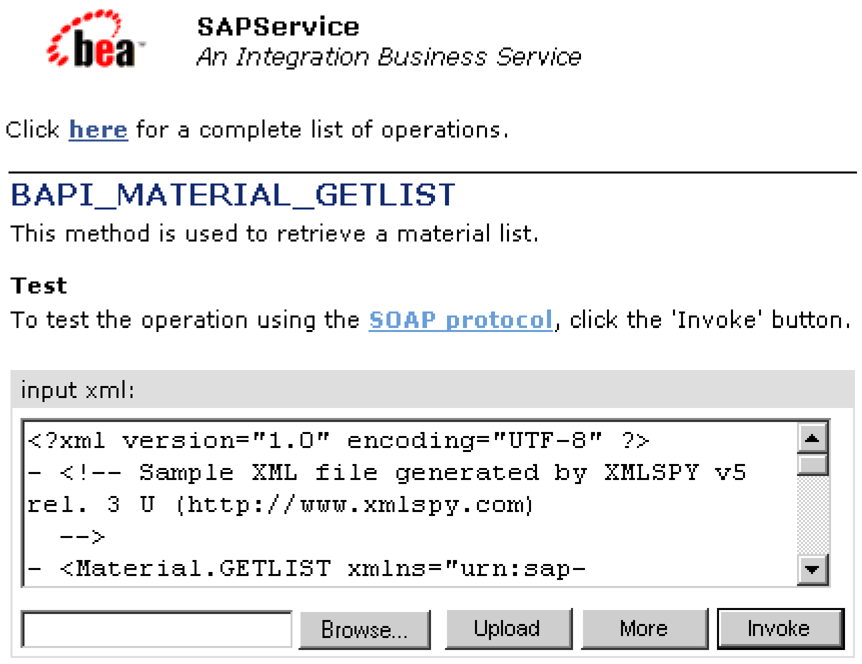
4. Click *OK*.

The business service and method appear below the Integration Business Services node.



In the left pane, all the available business services that were created appear. The SAPService node is expanded, and the BAPI_MATERIAL_GETLIST method automatically is selected.

On the right, the test pane for the BAPI_MATERIAL_GETLIST method opens.



SAPService
An Integration Business Service

Click here for a complete list of operations.

BAPI_MATERIAL_GETLIST

This method is used to retrieve a material list.

Test

To test the operation using the SOAP protocol, click the 'Invoke' button.

input xml:

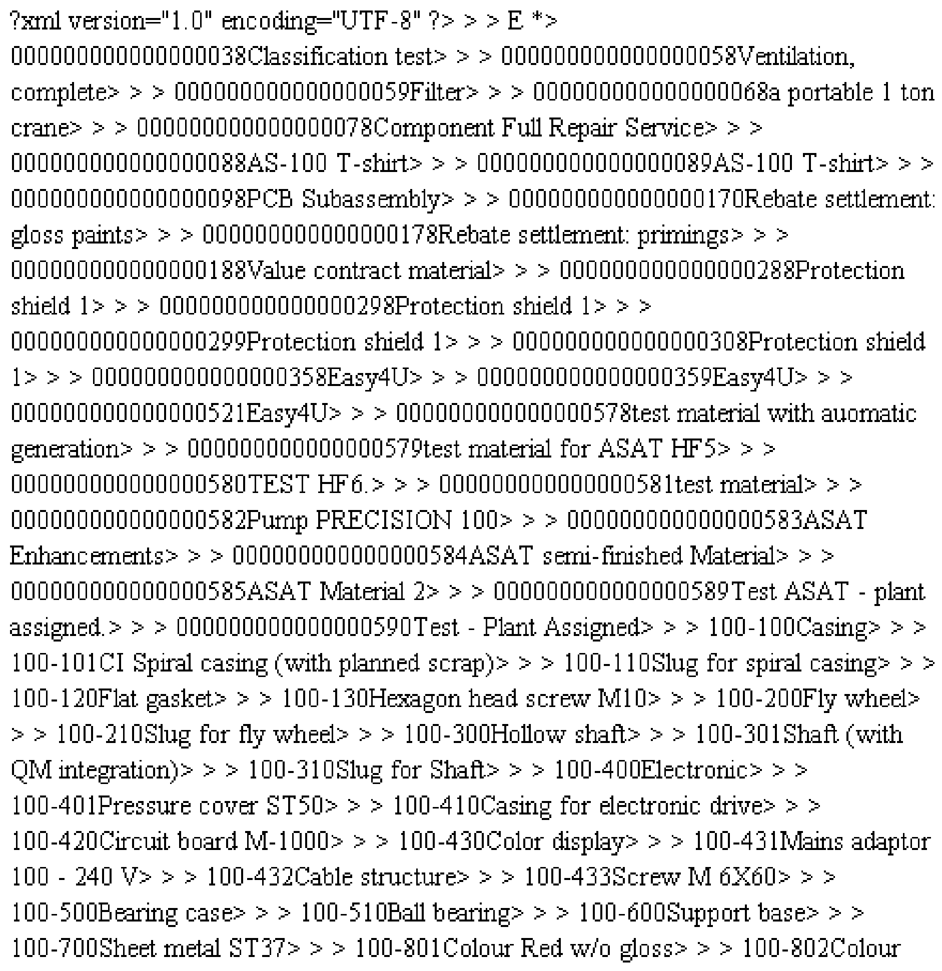
```
<?xml version="1.0" encoding="UTF-8" ?>
- <!-- Sample XML file generated by XMLSPY v5
rel. 3 U (http://www.xmlspy.com)
  -->
- <Material.GETLIST xmlns="urn:sap-
```

5. To invoke the service, enter a sample XML document in the input xml field.

For sample input XML, see *Retrieving a List of Materials Using the SAP BAPI_MATERIAL_GETLIST Method* on page A-24.

6. Click *Invoke*.

   The result appears in the right pane.



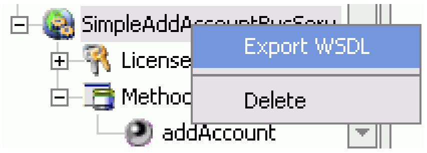
```
?xml version="1.0" encoding="UTF-8" ?> > > E *>
000000000000000038Classification test> > > 000000000000000058Ventilation,
complete> > > 000000000000000059Filter> > > 000000000000000068a portable 1 ton
crane> > > 000000000000000078Component Full Repair Service> > >
000000000000000088AS-100 T-shirt> > > 000000000000000089AS-100 T-shirt> > >
000000000000000098PCB Subassembly> > > 000000000000000170Rebate settlement:
gloss paints> > > 000000000000000178Rebate settlement: primings> > >
000000000000000188Value contract material> > > 000000000000000288Protection
shield 1> > > 000000000000000298Protection shield 1> > >
000000000000000299Protection shield 1> > > 000000000000000308Protection shield
1> > > 000000000000000358Easy4U> > > 000000000000000359Easy4U> > >
000000000000000521Easy4U> > > 000000000000000578test material with auomatic
generation> > > 000000000000000579test material for ASAT HF5> > >
000000000000000580TEST HF6.> > > 000000000000000581test material> > >
000000000000000582Pump PRECISION 100> > > 000000000000000583ASAT
Enhancements> > > 000000000000000584ASAT semi-finished Material> > >
000000000000000585ASAT Material 2> > > 000000000000000589Test ASAT - plant
assigned.> > > 000000000000000590Test - Plant Assigned> > > 100-100Casing> > >
100-101CI Spiral casing (with planned scrap)> > > 100-110Slug for spiral casing> > >
100-120Flat gasket> > > 100-130Hexagon head screw M10> > > 100-200Fly wheel>
> > 100-210Slug for fly wheel> > > 100-300Hollow shaft> > > 100-301Shaft (with
QM integration)> > > 100-310Slug for Shaft> > > 100-400Electronic> > >
100-401Pressure cover ST50> > > 100-410Casing for electronic drive> > >
100-420Circuit board M-1000> > > 100-430Color display> > > 100-431Mains adaptor
100 - 240 V> > > 100-432Cable structure> > > 100-433Screw M 6X60> > >
100-500Bearing case> > > 100-510Ball bearing> > > 100-600Support base> > >
100-700Sheet metal ST37> > > 100-801Colour Red w/o gloss> > > 100-802Colour
```

Exporting WSDL for Use in BEA WebLogic WorkShop Workflows

Because Application Explorer runs within BEA WebLogic Workshop, you can easily incorporate Web services into BEA WebLogic Workflows. To enable BEA WebLogic Workshop to use Web services, you simply export the WSDL to a directory accessible to BEA WebLogic Workshop.

Procedure How to Export WSDL for Use in BEA WebLogic Workshop Workflows

To export WSDL to a directory accessible to BEA WebLogic Workshop:

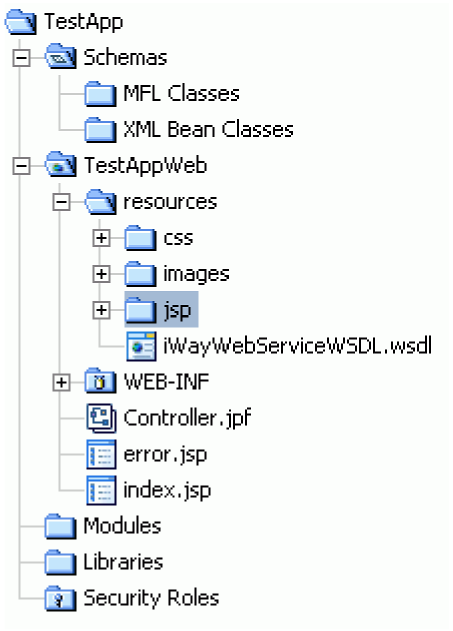


1. After you create the Web service, right-click the Web service name and select *Export WSDL*.

   The Save dialog box appears.

2. Save the WSDL to a directory that is accessible to BEA WebLogic Workshop, for example, the \resources directory in your BEA WebLogic Workshop Web application directory structure.

   The WSDL file appears under the resources folder of your Web application.



Identity Propagation

If you test or execute a Web service using a third party XML editor, for example XMLSPY, the Username and Password values that you specify in the SOAP header must be valid and are used to connect to SAP. The user name and password values that you provided for SAP during target creation using Application Explorer are overwritten for this Web service request. The following is a sample SOAP header that is included in the WSDL file for a Web service:

```
<SOAP-ENV:Header>
    <m:ibsinfo xmlns:m="urn:schemas-iwaysoftware-com:iwse">
        <m:service>String</m:service>
        <m:method>String</m:method>
        <m:license>String</m:license>
        <m:disposition>String</m:disposition>
        <m:Username>String</m:Username>
        <m:Password>String</m:Password>
        <m:language>String</m:language>
    </m:ibsinfo>
</SOAP-ENV:Header>
```

Note: You can remove the following tags from the SOAP header, since they are not required:

```
<m:disposition>String</m:disposition>
```

```
<m:language>String</m:language>
```

Example Retrieving a List of Materials Using the SAP BAPI_MATERIAL_GETLIST Method

The following input XML retrieves a list of materials using the SAP BAPI_MATERIAL_GETLIST method. You can use the input XML to test the Integration Business Service you created.

```
  <?xml version="1.0" encoding="UTF-8" ?>
- <!-- Sample XML file generated by XMLSPY v5 rel. 3 U
(http://www.xmlspy.com)
  -->
- <Material.GETLIST xmlns="urn:sap-com:document:sap:business"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="urn:sap-com:document:sap:business
C:\PROGRA~1\BEASYS~1\BEAAPP~1\sessions\default\SAP\beasap46\service_BAPI_
MATERIAL_GETLIST.xsd">
  <MAXROWS>1000</MAXROWS>
- <DISTRIBUTIONCHANNELSELECTION>
- <item>
  <SIGN />
  <OPTION />
  <DISTR_CHAN_LOW />
  <DISTR_CHAN_HIGH />
  </item>
  </DISTRIBUTIONCHANNELSELECTION>
- <MANUFACTURERPARTNUMB>
- <item>
  <MANU_MAT />
  <MFR_NO />
  </item>
  </MANUFACTURERPARTNUMB>
- <MATERIALSHORTDESCSEL>
- <item>
  <SIGN />
  <OPTION />
  <DESCR_LOW />
  <DESCR_HIGH />
  </item>
  </MATERIALSHORTDESCSEL>
- <MATNRLIST>
- <item>
  <MATERIAL />
  <MATL_DESC />
  <MATERIAL_EXTERNAL />
  <MATERIAL_GUID />
  <MATERIAL_VERSION />
  </item>
  </MATNRLIST>
- <MATNRSELECTION>
- <item>
  <SIGN>E</SIGN>
  <OPTION>BT</OPTION>
  <MATNR_LOW>1000</MATNR_LOW>
  <MATNR_HIGH>1010</MATNR_HIGH>
  </item>
  </MATNRSELECTION>
- <PLANTSELECTION>
- <item>
  <SIGN />
  <OPTION />
  <PLANT_LOW />
  <PLANT_HIGH />
  </item>
  </PLANTSELECTION>
- <RETURN>
- <item>
  <TYPE />
  <ID />
  <NUMBER />
  <MESSAGE />
  <LOG_NO />
  <LOG_MSG_NO />
  <MESSAGE_V1 />
  <MESSAGE_V2 />
  <MESSAGE_V3 />
  <MESSAGE_V4 />
  <PARAMETER />
  <ROW>0</ROW>
  <FIELD />
  <SYSTEM />
  </item>
  </RETURN>
- <SALESORGANISATIONSELECTION>
- <item>
  <SIGN />
  <OPTION />
  <SALESORG_LOW />
  <SALESORG_HIGH />
  </item>
  </SALESORGANISATIONSELECTION>
- <STORAGELOCATIONSELECT>
- <item>
  <SIGN />
  <OPTION />
  <STLOC_LOW />
  <STLOC_HIGH />
  </item>
  </STORAGELOCATIONSELECT>
  </Material.GETLIST>
```

Adding a Control for a Resource in BEA WebLogic Workshop

Java controls provide a convenient way to incorporate access to resources. You can add controls in BEA WebLogic Workshop to use Web services created by Application Explorer, or you can add controls that enable you to take advantage of the JCA resources of Application Explorer.

Adding a Web Service Control to a BEA WebLogic Workshop Application

After you create a Web service using Application Explorer and export the WSDL file, you can create a control for the Web service.

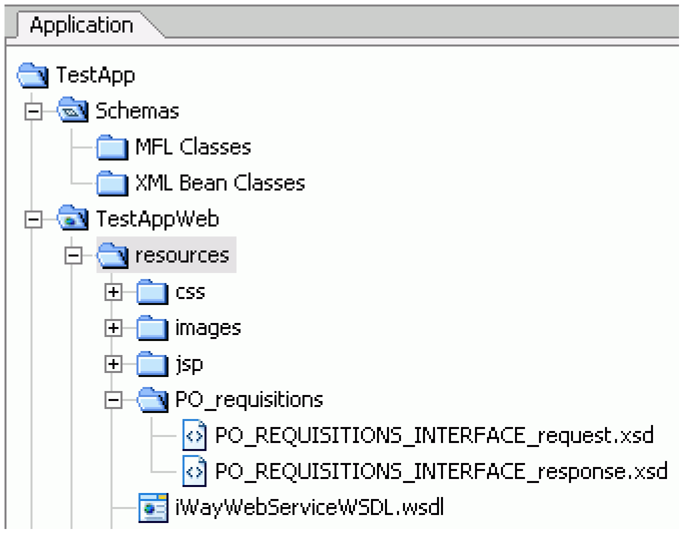
For more information on exporting a WSDL file, see *How to Export WSDL for Use in BEA WebLogic Workshop Workflows* on page A-22.

Procedure How to Add a Web Service Control

To add a Web service control:

1. After exporting the WSDL file from Application Explorer, locate the file in the Application tab of your BEA WebLogic Workshop application.

   For example, a WSDL file saved to the \resources directory in your BEA WebLogic Workshop Web application directory structure appears as follows.



2. Right-click the *WSDL* file and select *Generate Service Control*.

   The control for the WSDL appears below the WSDL file in the resources tree.

Extensible CCI Control

The following section describes the enhanced CCI control, which is extensible and provides JCX with typed inputs and outputs for JCA in BEA WebLogic Workshop.

Overview

The extensible CCI control now offers:

- **Method and tag validation.** BEA WebLogic Workshop provides warnings regarding invalid methods and tags.
- **Improved error handling.**

You can now define new methods that rely on the generic *service* and *authService* methods. For example, you can define a JCX with a new method such as the following, without having to write casting code or explicit transformations:

```
sapComDocumentSapRfcFunctions.BAPIMATERIALGETDETAILResponseDocument
getDetail(sapComDocumentSapRfcFunctions.BAPIMATERIALGETDETAILDocument
aRequest) throws java.lang.Exception
```

In addition, the extensible CCI control now generates a JCX file to which you can add your own methods.

Using the Extensible CCI Control

The extensible CCI control functions much like a database control since it generates JCX files to which you can add your own methods.

Your own methods can use the correct input and output types rather than the generic XmlObject types that the JCA control uses. Since the control is just a proxy that uses a reflection to call the relevant method, it will take care of the casting for you. There is no longer a need to write custom code that does the cast or transformations that are cast between an XmlObject.

For example, instead of the generic XmlObject:

```
XmlObject service(XmlObject input) throws java.lang.Exception;
```

you will be calling:

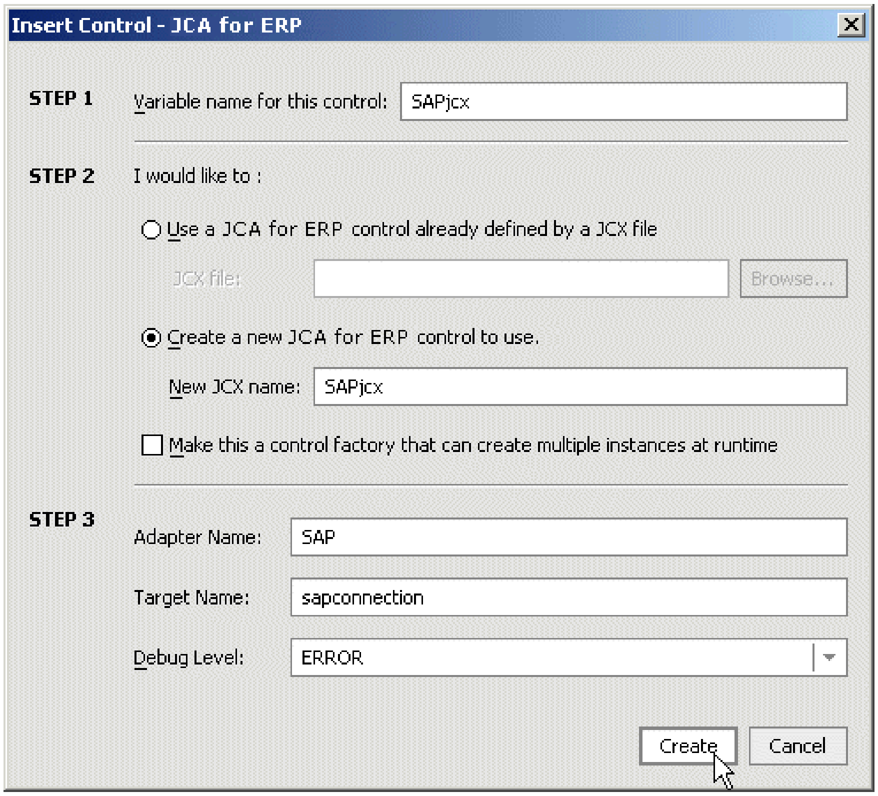
```
BAPIMATERIALGETDETAILResponseDocument
getDetail(BAPIMATERIALGETDETAILDocument aRequest) throws
java.lang.Exception;
```

Example Defining a Control Using the Extensible CCI Control

The following sample JCX demonstrates how to define a control that uses the SAP BAPI_MATERIAL_GET_DATA method using the extensible CCI control in BEA WebLogic Workshop.

1. Start BEA WebLogic Workshop and create a new project.
2. Click *Integration Controls* and select *JCA for ERP.*

   The Insert Control - JCA for ERP dialog box opens.



Insert Control - JCA for ERP

STEP 1 Variable name for this control: SAPjcx

STEP 2 I would like to :

( ) Use a JCA for ERP control already defined by a JCX file

JCX file:

(●) Create a new JCA for ERP control to use.

New JCX name: SAPjcx

[ ] Make this a control factory that can create multiple instances at runtime

STEP 3 Adapter Name: SAP

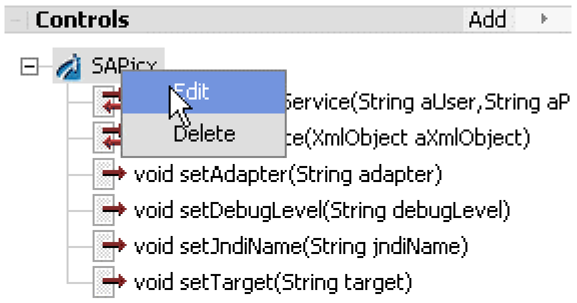
Target Name: sapconnection

Debug Level: ERROR

3. Perform the following steps:
   a. Provide a variable name for this control.
   b. Click *Create a new JCA for ERP control* to use and provide a new JCX name.
   c. Enter the adapter name, target name, and select a debug level from the drop-down list.
4. Click *Create.*

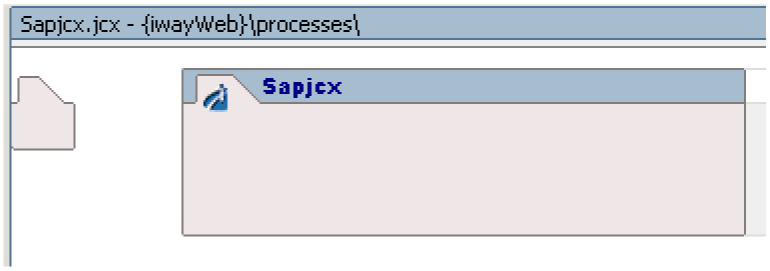
A new JCX file is created.

To edit an existing control, right click the control and select *Edit*.

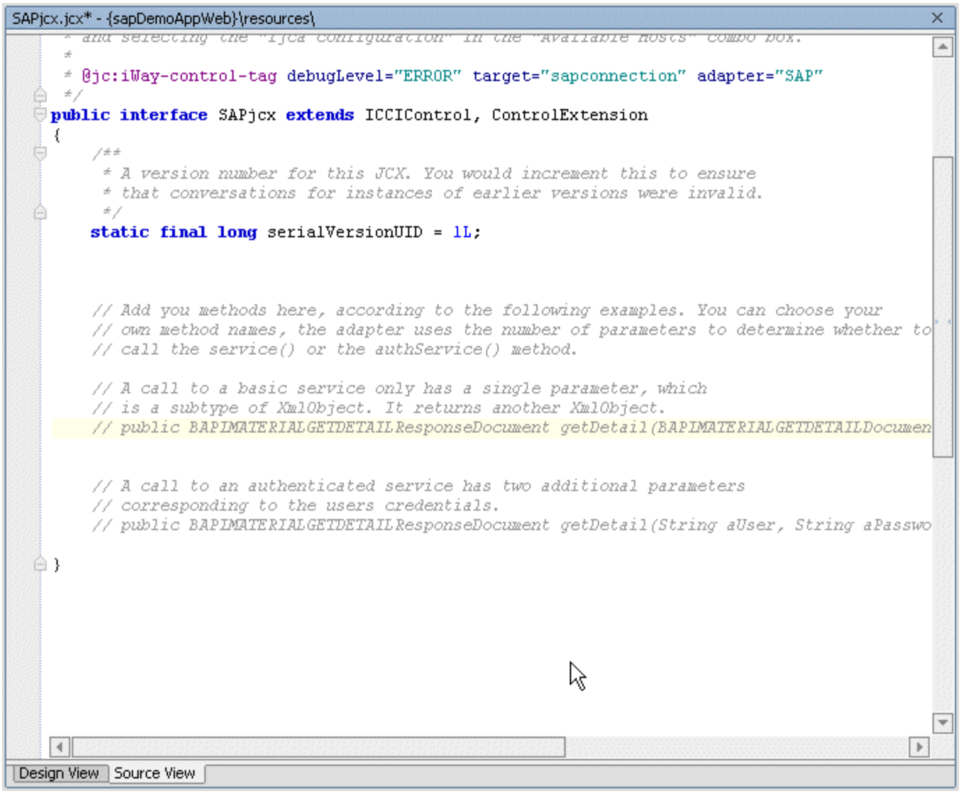


```
...Service(String aUser,String aP
...e(XmlObject aXmlObject)
void setAdapter(String adapter)
void setDebugLevel(String debugLevel)
void setJndiName(String jndiName)
void setTarget(String target)
```

The Design view is displayed.



5. Click *Source View*.



SAPjcx.jcx* - {sapDemoAppWeb}\resources\

```
 * and selecting the "ijca configuration" in the "Available Hosts" combo box.
 *
 * @jc:iWay-control-tag debugLevel="ERROR" target="sapconnection" adapter="SAP"
 */
public interface SAPjcx extends ICCIControl, ControlExtension
{
    /**
     * A version number for this JCX. You would increment this to ensure
     * that conversations for instances of earlier versions were invalid.
     */
    static final long serialVersionUID = 1L;



    // Add you methods here, according to the following examples. You can choose your
    // own method names, the adapter uses the number of parameters to determine whether to
    // call the service() or the authService() method.

    // A call to a basic service only has a single parameter, which
    // is a subtype of XmlObject. It returns another XmlObject.
    // public BAPIMATERIALGETDETAILResponseDocument getDetail(BAPIMATERIALGETDETAILDocumen


    // A call to an authenticated service has two additional parameters
    // corresponding to the users credentials.
    // public BAPIMATERIALGETDETAILResponseDocument getDetail(String aUser, String aPasswo

}
```

You can add your own methods that call the adapter's services.

APPENDIX B

Using Application Explorer in BEA WebLogic Workshop for Event Handling

Topics:

This section describes how to use Java Swing Application Explorer running in BEA WebLogic Workshop to create events for SAP. In addition, this section provides information on using events in a clustered BEA WebLogic environment.

Starting Application Explorer in BEA WebLogic WorkShop

The server must be started where Application Explorer is running. Before you can use Application Explorer, you must start BEA WebLogic server.

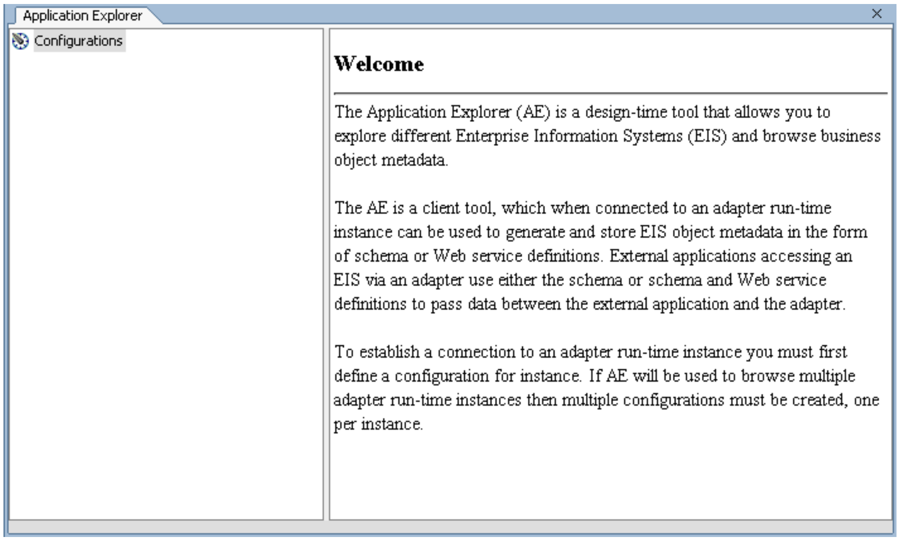
You can run Application Explorer in BEA WebLogic Workshop using an Integration Business Services Engine (iBSE) configuration. If you want to use Application Explorer with a JCA configuration instead of iBSE, you must use the servlet version of Application Explorer that runs outside of WebLogic Workshop. For more information about the servlet version, see Chapter 3, *Creating XML Schemas for SAP*.

Procedure How to Start Application Explorer

To start Application Explorer running in BEA WebLogic Workshop:

1. Ensure the server on which Application Explorer is deployed is started.
2. Start BEA WebLogic Workshop.
3. From the BEA WebLogic Workshop View menu, select *Windows* and then, *Application Explorer.*

   Application Explorer opens in BEA WebLogic Workshop.



Welcome

The Application Explorer (AE) is a design-time tool that allows you to explore different Enterprise Information Systems (EIS) and browse business object metadata.

The AE is a client tool, which when connected to an adapter run-time instance can be used to generate and store EIS object metadata in the form of schema or Web service definitions. External applications accessing an EIS via an adapter use either the schema or schema and Web service definitions to pass data between the external application and the adapter.

To establish a connection to an adapter run-time instance you must first define a configuration for instance. If AE will be used to browse multiple adapter run-time instances then multiple configurations must be created, one per instance.

You can resize and drag-and-drop the Application Explorer window within BEA WebLogic Workshop. For example, you can drag it to the upper part of BEA WebLogic Workshop.

Understanding Event Functionality

Events are generated as a result of activity in an application system. You can an use an event to trigger an action in your application. For example, SAP R/3 may generate an event when customer information is updated. If your application must perform when this happens, your application is a consumer of this event.

After you create a connection to your application system, you can add events using Java Swing Application Explorer. To create an event, you must create a port and a channel.

- Port

  A port associates a particular business object exposed by the adapter with a particular disposition. A disposition defines the protocol and location of the event data. The port defines the end point of the event consumption. For more information, see *Creating an Event Port*.

- Channel

  A channel represents configured connections to particular instances of back-end systems. A channel binds one or more event ports to a particular listener managed by the adapter. For more information, see *Creating a Channel* on page B-18.

Creating an Event Port

The following procedures describe how to create an event port using Java Swing Application Explorer. When you use the Application Explorer with an Integration Business Services Engine (iBSE) implementation, the following port dispositions are available:

- File
- iBSE
- MSMQ
- JMS queue
- SOAP
- HTTP
- MQ Series
- MAIL

Note: The MAIL disposition option will be supported in a future release.

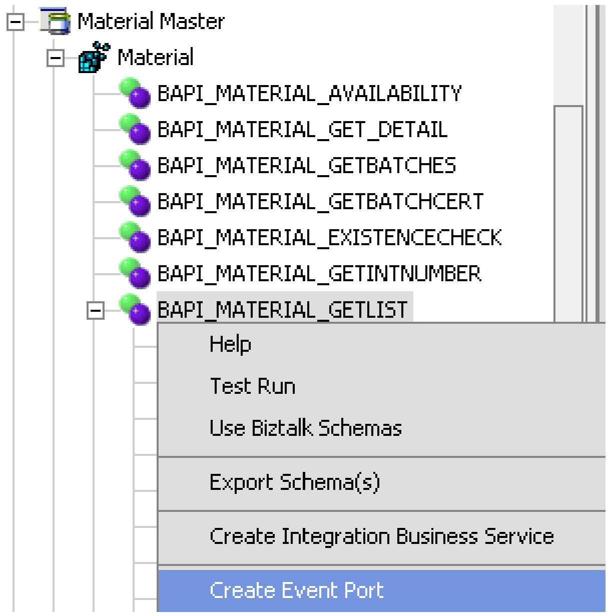
The following dispositions are available when using Application Explorer in conjunction with a JCA connector implementation.

- File

- HTTP
- JMS queue
- MQ Series

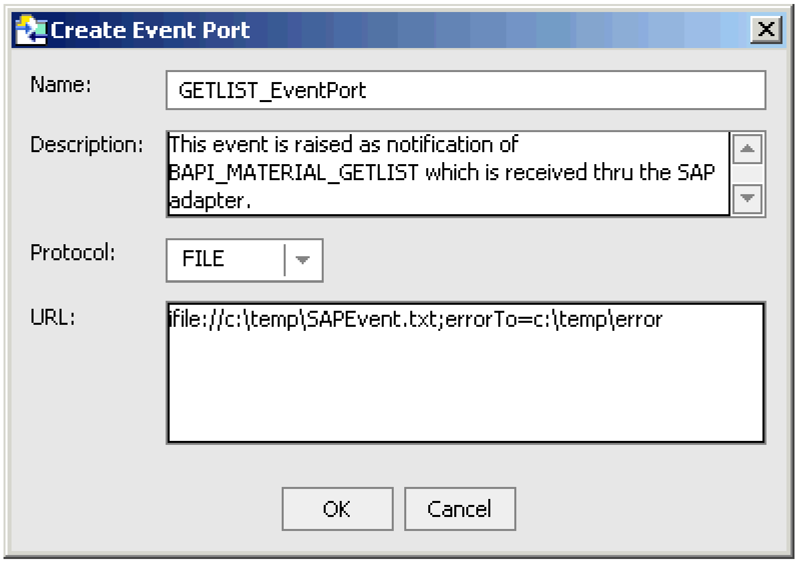
Procedure How to Create an Event Port for the File Disposition

To create an event port for the File disposition:



1. In the Business Object Repository, right-click the *BAPI_MATERIAL_GETLIST* method and select *Create Event Port*.

The Create Event Port dialog box opens.



Name: GETLIST_EventPort

Description: This event is raised as notification of BAPI_MATERIAL_GETLIST which is received thru the SAP adapter.

Protocol: FILE

URL: file://c:\temp\SAPEvent.txt;errorTo=c:\temp\error

a. In the Name field, type a name for the event port, for example, GETLIST_EventPort.

b. In the Description field, type a brief description.

c. From the Protocol drop-down list, select *FILE*.

d. In the URL field, type a destination where the event data is written using the following format:

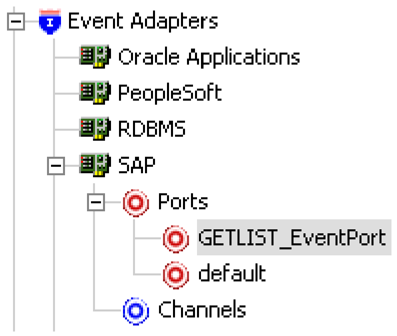
```
file://location;[errorTo=errorDest]
```

The following table describes the disposition parameters.

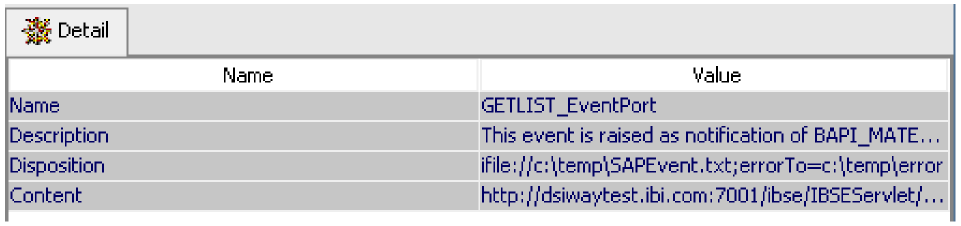
Parameter	**Description**
location	The full directory path and file name to which the data is written.
errorDest	Location where error logs are sent. Optional. A predefined port name or another disposition URL. The URL must be complete, including the protocol.

2. Click *OK*.

In the left pane, the event port you created appears below the Ports node.



In the right pane, a table appears that summarizes the information associated with the event port you created.



Detail

Name	Value
Name	GETLIST_EventPort
Description	This event is raised as notification of BAPI_MATE...
Disposition	ifile://c:\temp\SAPEvent.txt;errorTo=c:\temp\error
Content	http://dsiwaytest.ibi.com:7001/ibse/IBSEServlet/...

You are ready to associate the event port for File with a channel.

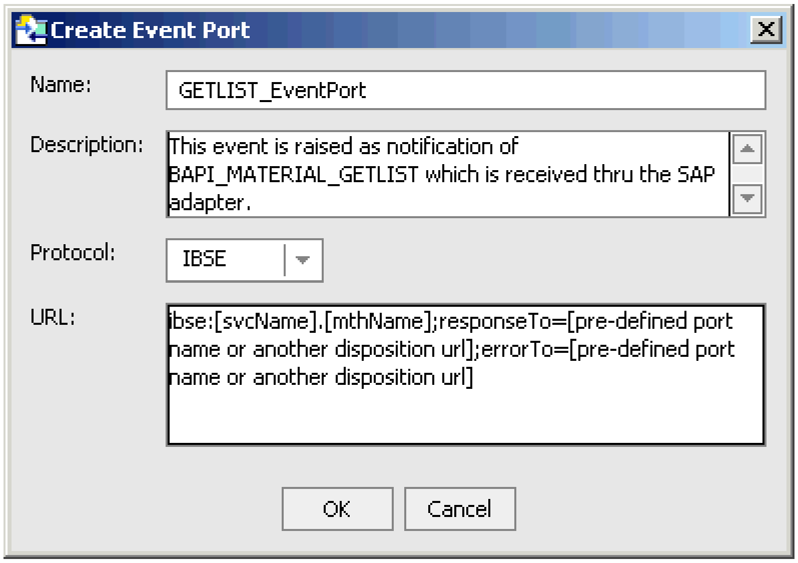
Procedure How to Create a Port for the iBSE Disposition

The iBSE disposition allows an event to launch an Integration Business Service method.

To create a port for iBSE:

1. From the Business Object Repository, right-click the *BAPI_MATERIAL_GETLIST* method and select *Create Event Port*.

The Create Event Port dialog box opens.



Name: GETLIST_EventPort

Description: This event is raised as notification of BAPI_MATERIAL_GETLIST which is received thru the SAP adapter.

Protocol: IBSE

URL: ibse:[svcName].[mthName];responseTo=[pre-defined port name or another disposition url];errorTo=[pre-defined port name or another disposition url]

a. In the Name field, type a name for the event port, for example, GETLIST_EventPort.

b. In the Description field, type a brief description.

c. From the Protocol drop-down list, select *IBSE*.

d. In the URL field, enter an iBSE destination using the following format:

```
ibse:/svcName.methName[;responseTo=respDest][;errorTo=errorDest]
```

The following table describes the disposition parameters.

Parameter	**Description**
svcName	Name of the service created with iBSE.
methName	Name of the method created for the Web service.
respDest	Location to post responses to the Web service. Optional. A predefined port name or another disposition URL. The URL must be complete, including the protocol.
errorDest	Location where error logs are sent. Optional. A predefined port name or another disposition URL. The URL must be complete, including the protocol.

2. Click *OK*.

The event port you created appears in the left pane under the Ports node.

In the right pane, a table appears that summarizes the information associated with the port you created. You are ready to associate the event port for IBSE with a channel.

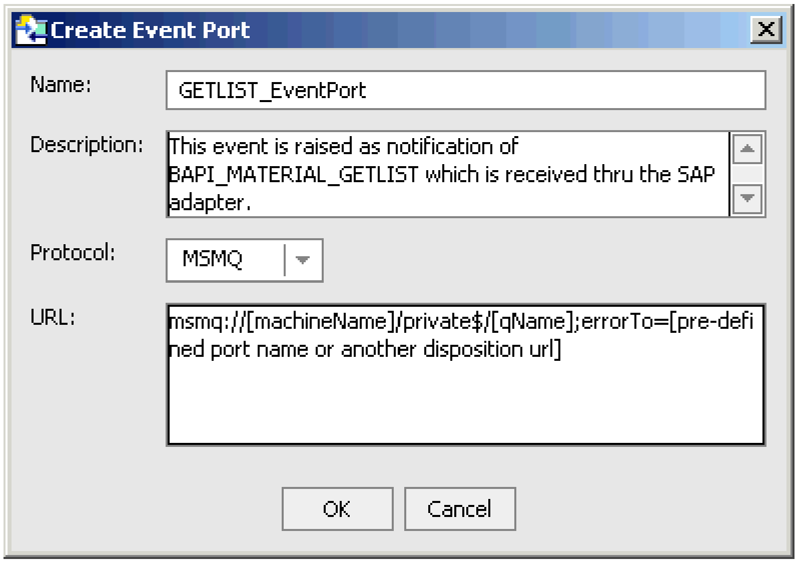
Procedure How to Create a Port for the MSMQ Disposition

The MSMQ disposition supports public and private queues.

To create a port for MSMQ:

1. In the Business Object Repository, right-click the *BAPI_MATERIAL_GETLIST* method and select *Create Event Port*.

   The Create Event Port dialog box opens.



Name: GETLIST_EventPort

Description: This event is raised as notification of BAPI_MATERIAL_GETLIST which is received thru the SAP adapter.

Protocol: MSMQ

URL: msmq://[machineName]/private$/[qName];errorTo=[pre-defined port name or another disposition url]

   a. In the Name field, type a name for the event port, for example, GETLIST_EventPort.
   b. In the Description field, type a brief description.
   c. From the Protocol drop-down list, select *MSMQ*.
   d. In the URL field, enter an MSMQ destination using the following format:

      `msmq://host/queueType/queueName[;errorTo=errorDest]`

      The following table defines the disposition parameters.

Parameter	**Description**
host	Name of the host on which the Microsoft Queuing system runs.

Parameter	**Description**
queueType	Type of queue. For private queues, enter *Private$*. Private queues are queues that are not published in Active Directory. They appear only on the local computer that contains them. Private queues are accessible only by Message Queuing applications that recognize the full path name or format name of the queue.
queueName	Name of the queue where messages are placed.
errorDest	Location where error logs are sent. Optional. A predefined port name or another disposition URL. The URL must be complete, including the protocol.

2. Click *OK*.

   In the left pane, the event port you created appears under the Ports node.

   In the right pane, a table appears that summarizes the information associated with the port you created. You are ready to associate the event port for MSMQ with a channel.

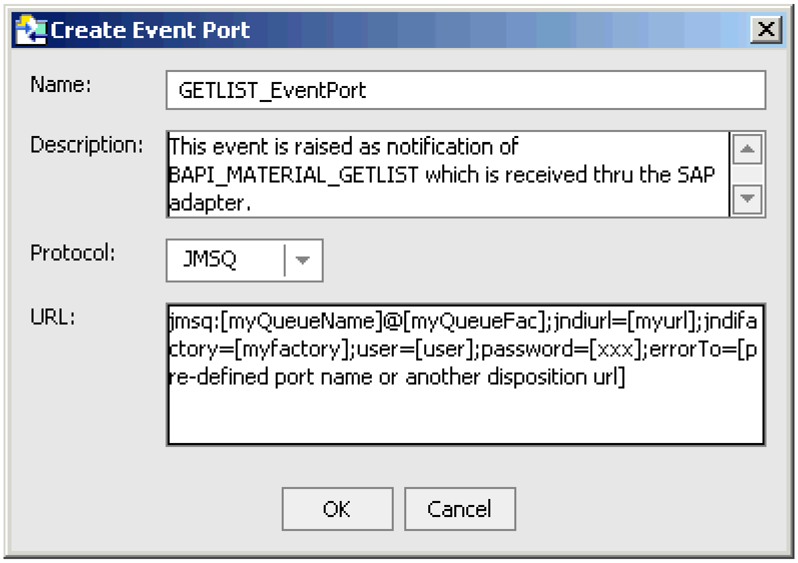
Procedure How to Create a Port for the JMS Disposition

The JMS disposition enables an event to be enqueued to a JMS queue.

To create a port for a JMS queue disposition:

1. In the Business Object Repository, right-click the *BAPI_MATERIAL_GETLIST* method and select *Create Event Port*.

   The Create Event Port dialog box opens.



Name: GETLIST_EventPort

Description: This event is raised as notification of BAPI_MATERIAL_GETLIST which is received thru the SAP adapter.

Protocol: JMSQ

URL: jmsq:[myQueueName]@[myQueueFac];jndiurl=[myurl];jndifactory=[myfactory];user=[user];password=[xxx];errorTo=[pre-defined port name or another disposition url]

a. In the Name field, type a name for the event port, for example, GETLIST_EventPort.

b. In the Description field, type a brief description.

c. From the Protocol drop-down list, select *JMSQ*.

d. In the URL field, enter an JMSQ destination using the following format:

```
jmsq:queue@conn_factory;jndiurl=jndi_url;jndifactory=jndi_factory;
user=userID;password=pass[;errorTo=errorDest]
```

The following table describes the disposition parameters.

Parameter	**Description**
queue	Name of a queue to which events are emitted.
conn_factory	The connection factory, a resource which contains information about the JMS Server. The BEA WebLogic connection factory is: `javax.jms.QueueConnectionFactory`

Parameter	Description
jndi_url	The URL of the application server. For BEA WebLogic Server this is `t3://host:port` where: *`host`* Is the machine name where BEA WebLogic Server resides. *`port`* Is the port on which BEA WebLogic Server is listening. The default port if not changed at installation is 7001.
jndi_factory	Is JNDI context.INITIAL_CONTEXT_FACTORY and is provided by the JNDI service provider. For BEA WebLogic Server, the BEA WebLogic factory is weblogic.jndi.WLInitialContextFactory.
userID	User ID associated with this queue.
pass	Password for this user ID.
errorDest	Location where error logs are sent. Optional. A predefined port name or another disposition URL. The URL must be complete, including the protocol.

2. Click *OK*.

   The event port you created appears in the left pane under the Ports node.

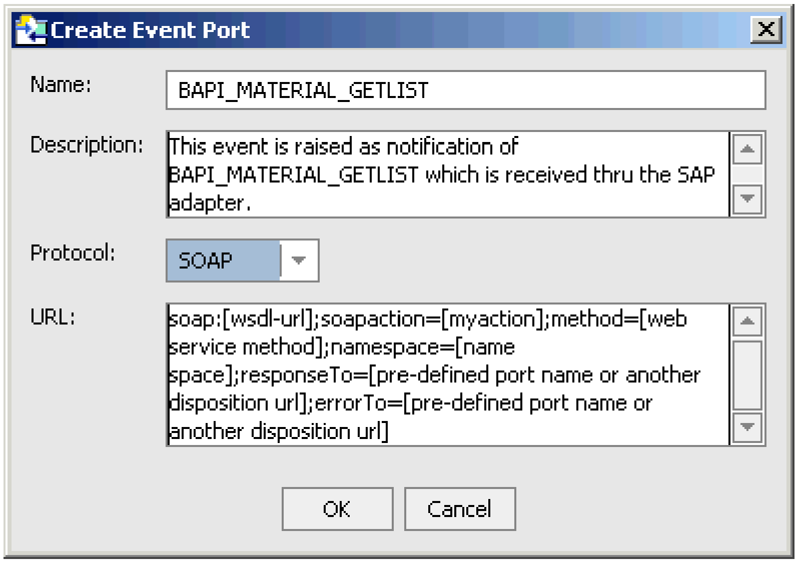
   In the right pane, a table appears that summarizes the information associated with the port you created. You are ready to associate the event port for JMSQ with a channel.

Procedure How to Create a Port for the SOAP Disposition

To create a port for the SOAP disposition:

1. In the Business Object Repository, right-click the *BAPI_MATERIAL_GETLIST* method and select *Create Event Port*.

   The Create Event Port dialog box opens.



Name: BAPI_MATERIAL_GETLIST

Description: This event is raised as notification of BAPI_MATERIAL_GETLIST which is received thru the SAP adapter.

Protocol: SOAP

URL: soap:[wsdl-url];soapaction=[myaction];method=[web service method];namespace=[name space];responseTo=[pre-defined port name or another disposition url];errorTo=[pre-defined port name or another disposition url]

a. In the Name field, type a name for the event port, for example, BAPI_MATERIAL_GETLIST.

b. In the Description field, type a brief description.

c. From the Protocol drop-down list, select *SOAP*.

d. In the URL field, enter a SOAP destination using the following format:

```
soap:[wsdl-url];soapaction=[myaction];method=[web service
method];namespace=[namespace];responseTo=[pre-defined port name or
another disposition URL];errorTo=[pre-defined port name or another
disposition url]
```

The following table describes the disposition parameters.

Parameter	**Description**

wsdl-url	The URL to the WSDL file that is required to create the SOAP message. For example:
	`http://localhost:7001/ibse/IBSEServlet/test/webservice.ibs?wsdl`
	where:
	`webservice`
	Is the name of the Web service you created using Application Explorer.
	You can also open the WSDL file in a third party XML editor (for example, XMLSPY) and view the SOAP request settings to find this value.

soapaction	The method that will be called by the SOAP disposition. This value can be found in the WSDL file.
method	The Web service method you are using. This value can be found in the WSDL file.
namespace	The XML namespace you are using. This value can be found in the WSDL file.
responseTo	The location to which responses are posted, which can be a predefined port name or another URL. Optional. The URL must be complete, including the protocol.
errorTo	The location to which error logs are sent. Optional. A predefined port name or another disposition URL. The URL must be complete, including the protocol.

2. Click *OK*.

The event port you created appears in the left pane under the Ports node.

In the right pane, a table appears that summarizes the information associated with the port you created. You are ready to associate the event port for SOAP with a channel.

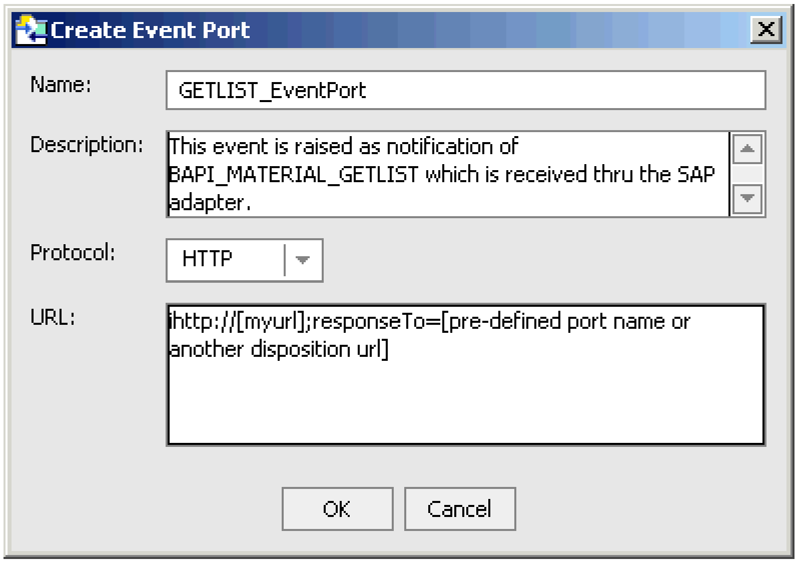
Procedure How to Create a Port for the HTTP Disposition

The HTTP disposition uses an HTTP URL to specify an HTTP end point to which the event document is posted.

To create a port for an HTTP disposition:

1. In the Business Object Repository, right-click the *BAPI_MATERIAL_GETLIST* method and select *Create Event Port*.

   The Create Event Port dialog box opens.



Name: GETLIST_EventPort

Description: This event is raised as notification of BAPI_MATERIAL_GETLIST which is received thru the SAP adapter.

Protocol: HTTP

URL: ihttp://[myurl];responseTo=[pre-defined port name or another disposition url]

   a. In the Name field, type a name for the event port, for example, GETLIST_EventPort.
   b. In the Description field, type a brief description.
   c. From the Protocol drop-down list, select *HTTP*.
   d. In the URL field, enter an HTTP destination using the following format:

      `ihttp://url;responseTo=respDest`

      The following table describes the disposition parameters.

Parameter	Description
url	The URL target for the post operation.

Parameter	**Description**
respDest	The location to which responses are posted. Optional. This can be a predefined port name or another disposition URL. The URL must be complete, including the protocol.
host	Name of the host on which the Web server resides.
port	Port number on which the Web server is listening.

2. Click *OK*.

   The event port you created appears in the left pane under the Ports node.

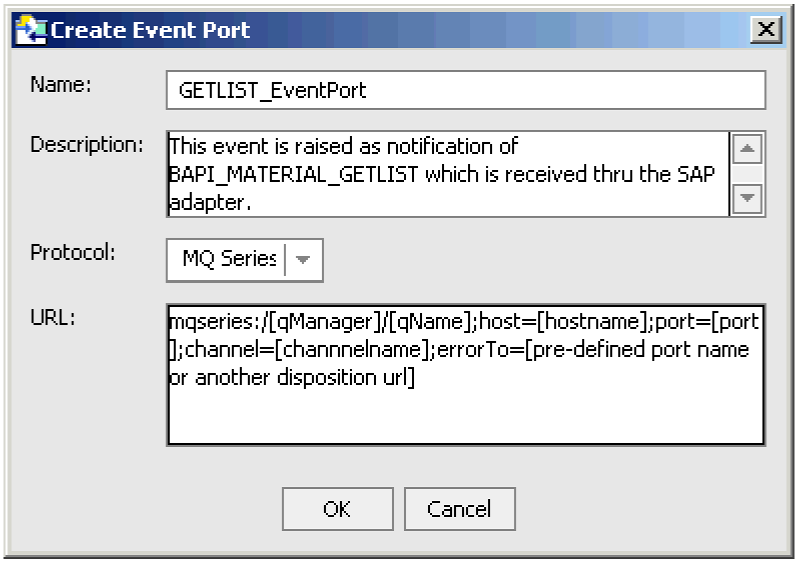
   In the right pane, a table appears that summarizes the information associated with the port you created. You are ready to associate the event port for HTTP with a channel.

Procedure How to Create a Port for the MQ Series Disposition

The MQ Series disposition allows an event to be enqueued to an MQ Series queue. Both queue manager and queue name may be specified. To create a port for MQ Series:

1. Select the *BAPI_MATERIAL_GETLIST* method from the Business Object Repository.
2. Right-click the method and select *Create Event Port*.

   The Create Event Port dialog box opens.



Name: GETLIST_EventPort

Description: This event is raised as notification of BAPI_MATERIAL_GETLIST which is received thru the SAP adapter.

Protocol: MQ Series

URL: mqseries:/[qManager]/[qName];host=[hostname];port=[port];channel=[channnelname];errorTo=[pre-defined port name or another disposition url]

   a. In the Name field, type a name for the event port, for example, GETLIST_EventPort.

b. In the Description field, type a brief description.

c. From the Protocol drop-down list, select *MQ Series*.

d. In the URL field, enter an MQ Series destination using the following format:

```
mqseries:/qManager/qName;host=hostName;port=portNum;
channel=chanName[;errorTo=errorDest]
```

The following table describes the disposition parameters.

Parameter	Description
qManager	Name of the queue manager to which the server must connect.
qName	Name of the queue where messages are placed.
hostName	Name of the host on which MQ Series resides (MQ client only).
portNum	Port number for connecting to an MQ Server queue manager (MQ client only).
chanName	Case-sensitive name of the channel that connects with the remote MQ Server queue manager (for MQ client only). The default MQ Series channel name is SYSTEM.DEF.SVRCONN.
errorDest	Location where error logs are sent. Optional. A predefined port name or another disposition URL. The URL must be complete, including the protocol.

3. Click *OK*.

In the left pane, the event port you created appears below the Ports node.

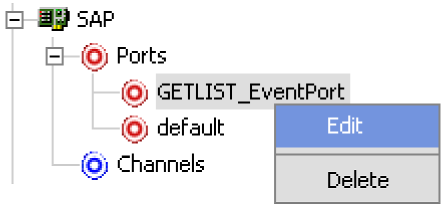
In the right pane, a table appears that summarizes the information associated with the port you created. You are ready to associate the event port for MQ Series with a channel.

Modifying an Event Port

The following procedures describe how to edit and delete an event port using Application Explorer. To review the port settings, select the port name. In the right pane, a table appears that summarizes the information associated with the event port you created.

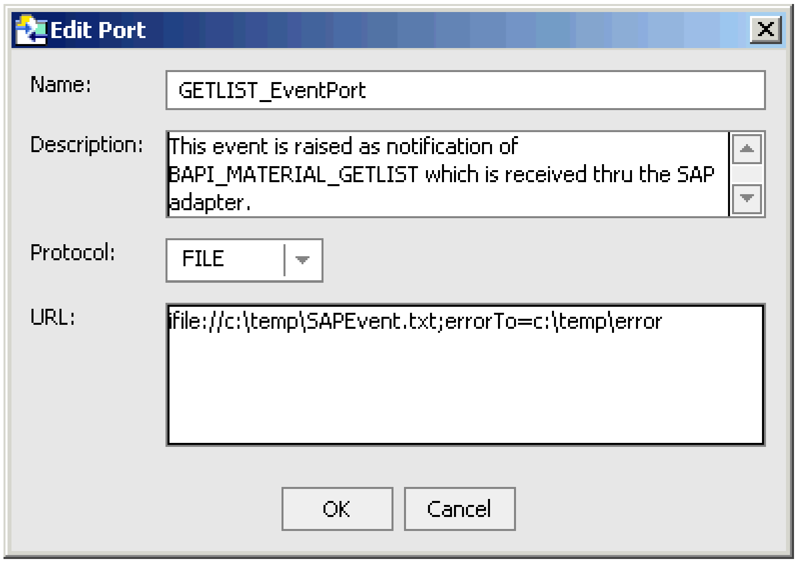
Procedure How to Edit an Event Port

To edit an existing event port:



1. To view the available ports, click the *Ports* node in the left pane.
2. Right-click the event port you want to edit and select *Edit*.

   The Edit Port dialog box opens.



Name: GETLIST_EventPort

Description: This event is raised as notification of BAPI_MATERIAL_GETLIST which is received thru the SAP adapter.

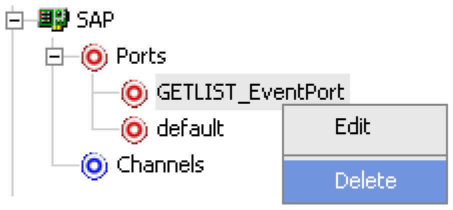
Protocol: FILE

URL: file://c:\temp\SAPEvent.txt;errorTo=c:\temp\error

3. Make the required changes and click *OK*.

Procedure How to Delete an Event Port

To delete an existing event port:



1. To view the available ports, click the *Ports* node in the left pane.

2. Right-click the event port you want to remove and select *Delete*.

   The event port disappears from the ports list in the left pane.

Creating a Channel

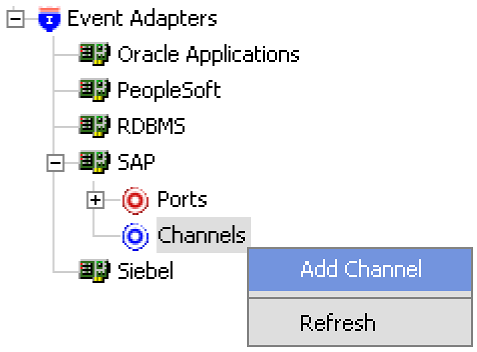
The following procedures describe how to create a channel to listen for SAP events. All defined event ports must be associated with a channel.

Procedure How to Create a Channel

To create a channel:

1. In the left pane, below the configuration you created (for example, SampleConfig), expand the *Event Adapters* node.

   The list of adapters appears.

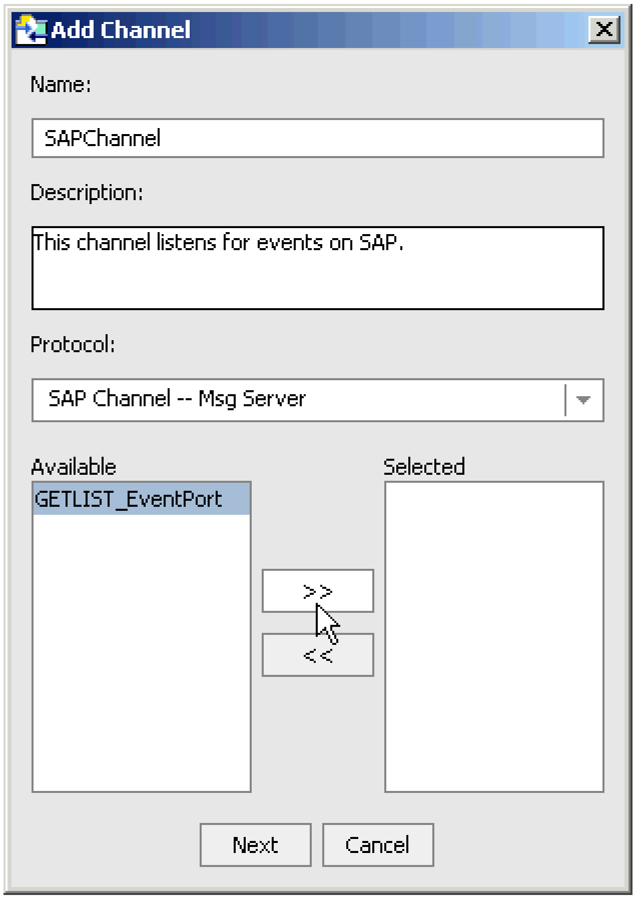


2. Click the *SAP* adapter node.

   The node expands, listing the Ports and Channels nodes.

3. Right-click *Channels* and select *Add Channel*.

The Add Channel dialog box opens.



Name: SAPChannel

Description: This channel listens for events on SAP.

Protocol: SAP Channel -- Msg Server

Available: GETLIST_EventPort

a. In the Name field, type a name for the channel, for example, SAPChannel.

b. In the Description field, type a brief description.

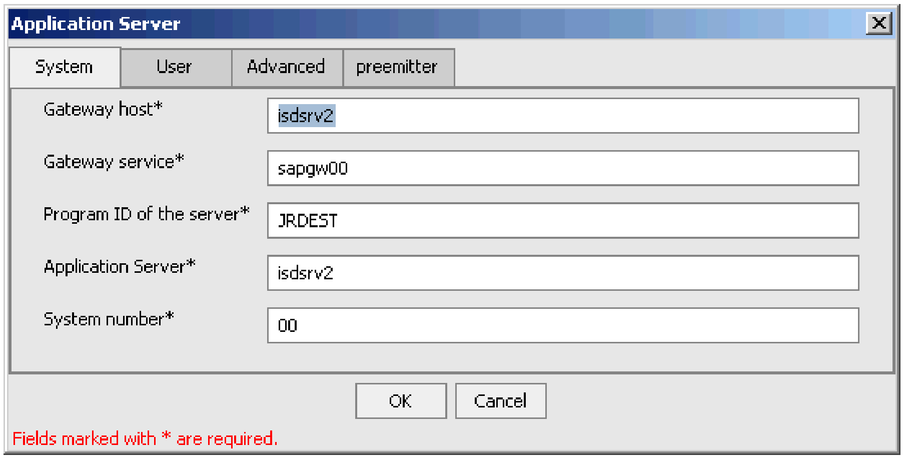
c. From the Protocol drop-down list, select *SAP Channel - App Server.*

d. To transfer an event port from the list of Available event ports to the list of Selected ports, click the double right (>>) arrow button.

Note: You can assign multiple event ports to a single channel.

4. Click *Next*.

The Application Server dialog box opens.



Gateway host* isdsrv2

Gateway service* sapgw00

Program ID of the server* JRDEST

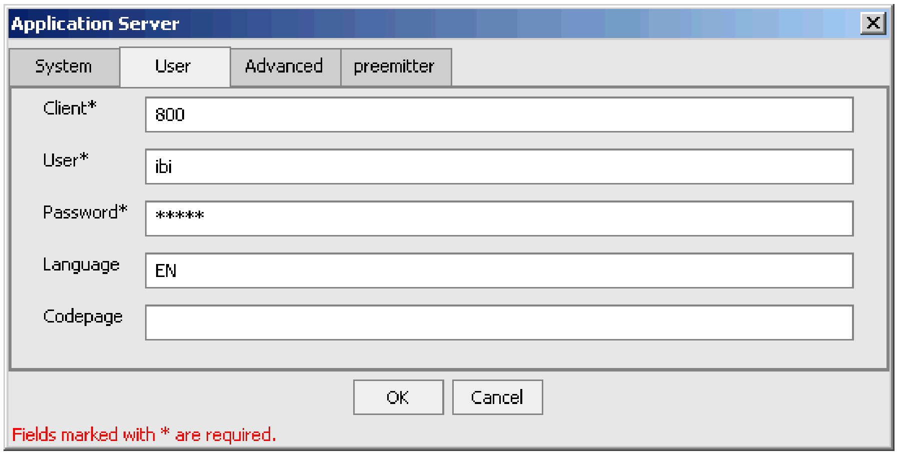
Application Server* isdsrv2

System number* 00

Fields marked with * are required.

a. On the System tab, type the system information that is specific to your SAP system.

b. Click the *User* tab.



Client* 800

User* ibi

Password* *****

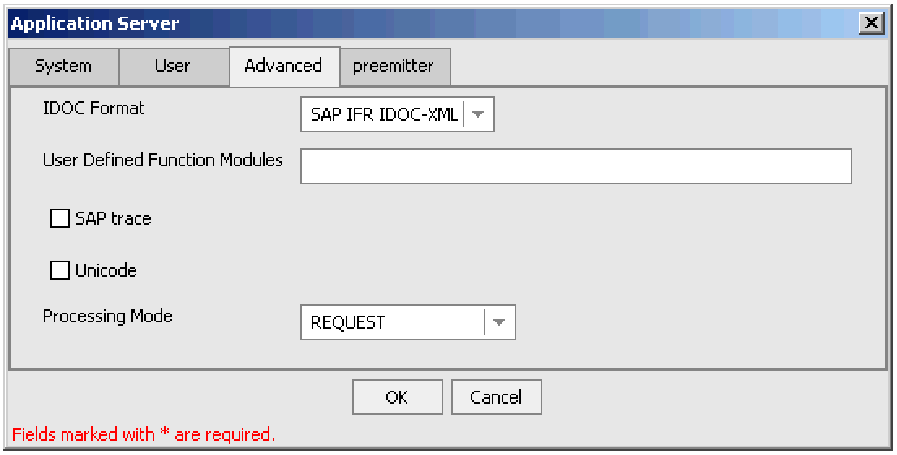
Language EN

Codepage

Fields marked with * are required.

c. Enter the user information that is specific to your SAP system.

d. Click the *Advanced* tab.



IDOC Format: SAP IFR IDOC-XML

User Defined Function Modules:

☐ SAP trace

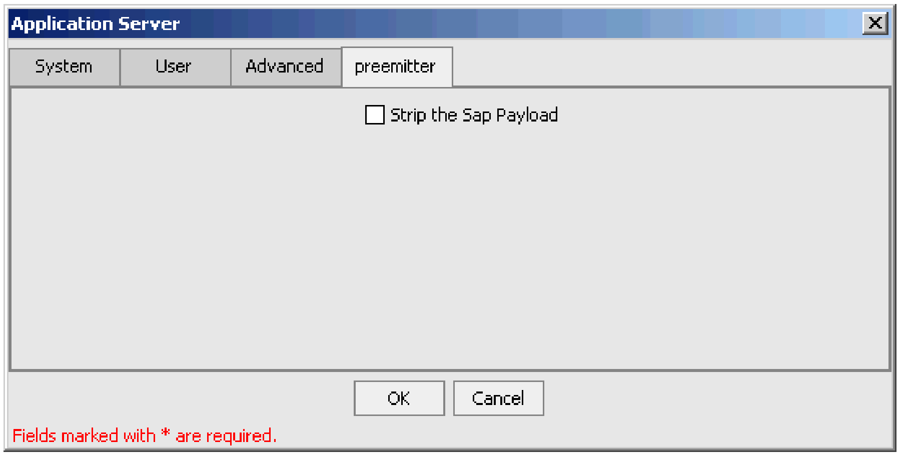
☐ Unicode

Processing Mode: REQUEST

Fields marked with * are required.

e. Specify any additional information or criteria for the channel you are creating.

f. Click the *preemitter* tab.



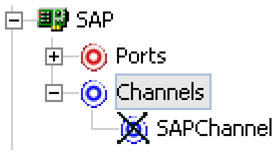
☐ Strip the Sap Payload

Fields marked with * are required.

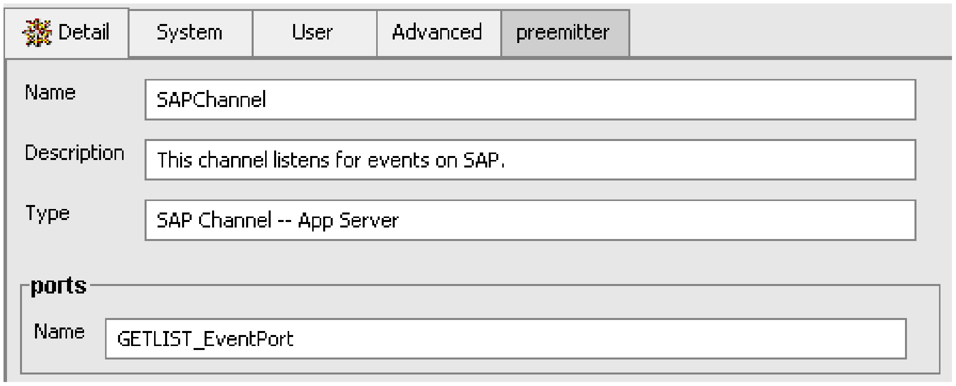
g. Click *Strip the Sap Payload* to strip the SAP payload of an event document.

5. Click *OK*.

The channel you created appears in the left pane below the Channels node.



When you select the channel, a table in the right pane summarizes all the information associated with the channel you created.



Name: SAPChannel

Description: This channel listens for events on SAP.

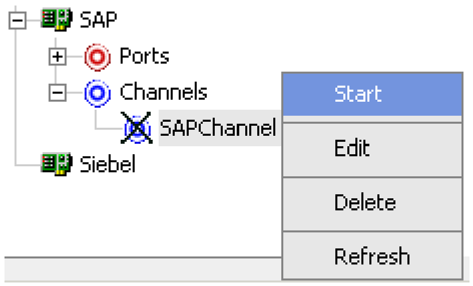
Type: SAP Channel -- App Server

ports

Name: GETLIST_EventPort

A Ports area appears on the Details tab that lists the event port you assigned to this channel.

You are ready to start your channel to listen for events.



6. In the left pane, right-click the channel, for example, SAPChannel, and select *Start*.

The channel you created is now active.

a. To stop the channel, right-click the channel.

b. Select *Stop*.

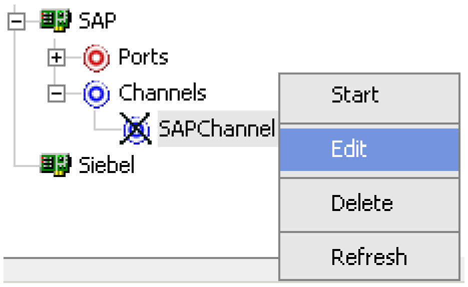
Modifying a Channel

The following procedures describe how to edit and delete a channel using Application Explorer. To review the channel settings, you select the channel name. In the right pane, a table appears that summarizes the information associated with the channel you created.

Procedure How to Edit a Channel

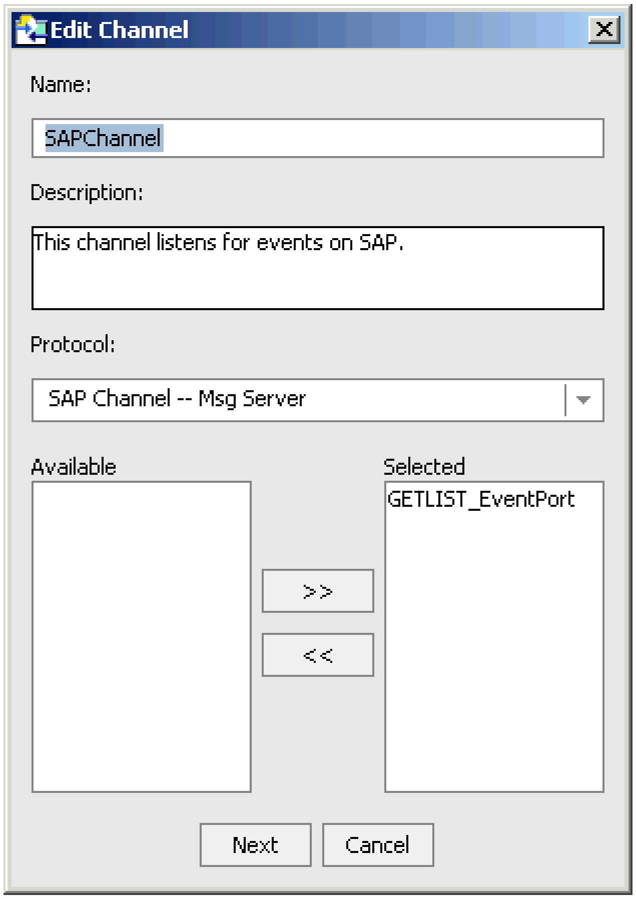
To edit an existing channel:

1. To view the available channels, click the Channels node in the left pane.



2. Right-click the channel you want to edit, for example, SAPChannel, and select *Edit*.

The Edit Channel dialog box opens.



Name:

SAPChannel

Description:

This channel listens for events on SAP.

Protocol:

SAP Channel -- Msg Server

Selected

GETLIST_EventPort

3. Make the required changes to the channel configuration.
4. Click *Next*.

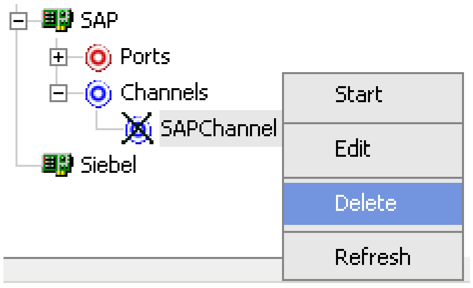
   The Message Server dialog box opens.

   a. If changes are required, click the appropriate tab (System, User, or Advanced).
   b. Make the required changes.
5. Click *OK*.

Procedure How to Delete a Channel

To delete an existing channel:

1. In the left pane, select the channel you want to delete.



2. Right-click the channel, for example, SAPChannel, and select *Delete*.

   The channel disappears from the Channels list.

Deploying Components in a Clustered BEA WebLogic Environment

Events can be configured in a clustered BEA WebLogic environment. You can deploy Integration Business Services Engine (iBSE) or JCA to this clustered environment.

A cluster consists of multiple server instances running simultaneously, yet appears to clients to be a single server instance. The server instances that contain a cluster can be run on one machine, but are usually run on multiple machines.

Clustering provides the following benefits:

- Load balancing
- High availability

Service requests are processed through the HTTP router and routed to an available managed server.

Events are server-specific and are not processed through the HTTP router. You must configure each server separately.

Procedure How to Deploy Integration Business Services Engine in a Clustered Environment

To deploy iBSE in a clustered environment:

1. Using the BEA Configuration Wizard:

   a. Configure an administrative server to manage the managed servers.

   b. Add and configure as many managed servers as required.

c. Add and configure an HTTP router. This does not have to be a part of WebLogic and can be an outside component.

d. If you configure the HTTP router within WebLogic, start it by entering the following command:

```
StartManagedWebLogic HTTPROUTER http://localhost:7001
```

where:

`HTTPROUTER`

Is the name of the server on which the HTTP router is running.

`http://`*`localhost`*`:7001`

Is the location of the admin console.

e. Add the managed servers to your cluster/clusters.

For more information on configuring WebLogic Integration for deployment in a clustered environment, see *Deploying WebLogic Integration Solutions*.

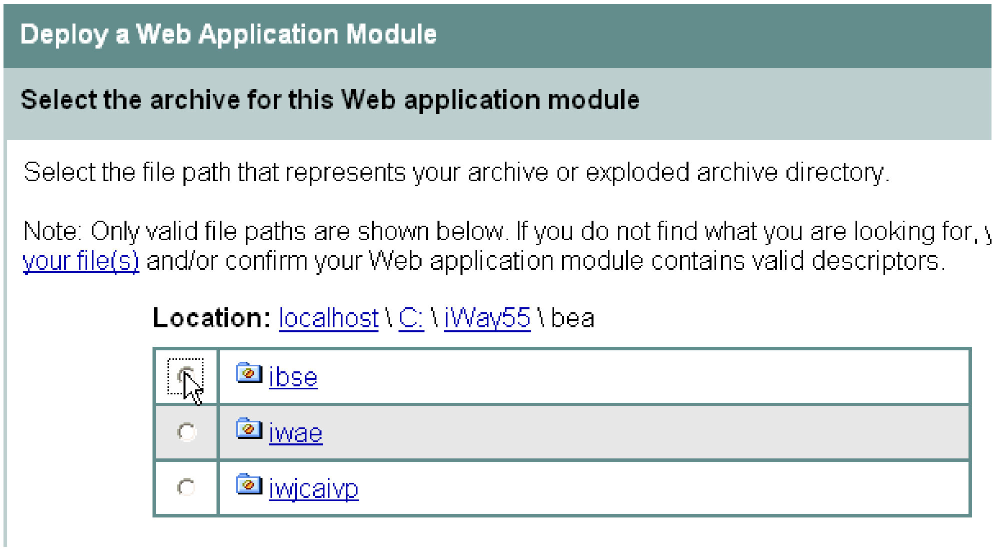
2. Start the WebLogic Server and open WebLogic Server Console.

3. Deploy iBSE to the cluster by selecting *Web Application Modules* from the Domain Configurations section, and clicking *Deploy a new Web Application Module*.

   A page appears for you to specify where the Web application is located.

   Note: You can deploy JCA to a cluster, but you can only point it to one directory, and to the machine on which it is installed.

4. To deploy iBSE, select the option button next to the ibse directory and then click *Target Module*.



Deploy a Web Application Module

Select the archive for this Web application module

Select the file path that represents your archive or exploded archive directory.

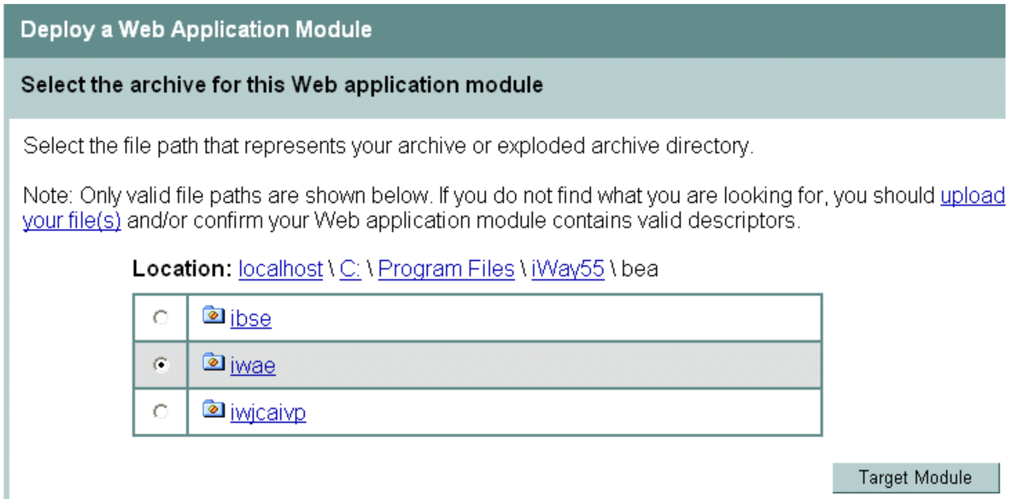
Note: Only valid file paths are shown below. If you do not find what you are looking for, y
your file(s) and/or confirm your Web application module contains valid descriptors.

Location: localhost \ C: \ iWay55 \ bea

○	ibse
○	iwae
○	iwjcaivp

5. To deploy servlet Application Explorer, select the option button next to the iwae directory and then click *Target Module*.

If you are using servlet Application Explorer, deploy it only on the admin server or one of the managed servers.



Deploy a Web Application Module

Select the archive for this Web application module

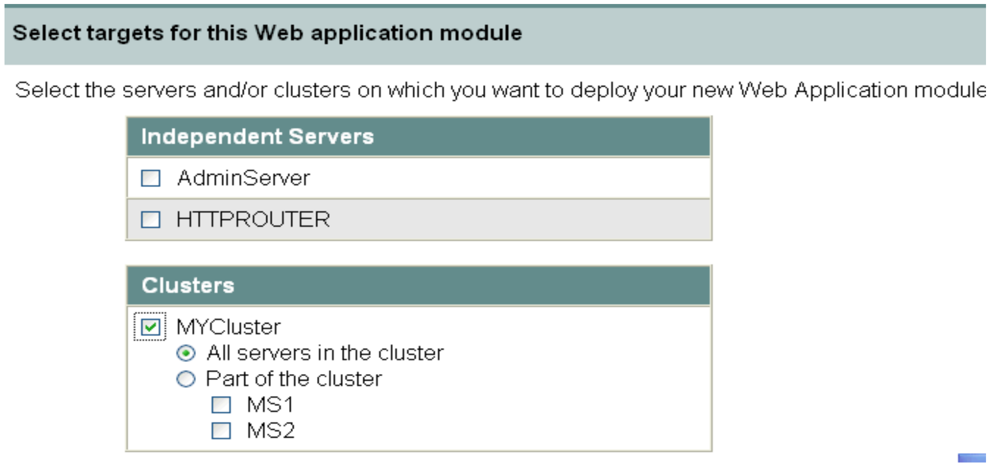
Select the file path that represents your archive or exploded archive directory.

Note: Only valid file paths are shown below. If you do not find what you are looking for, you should upload your file(s) and/or confirm your Web application module contains valid descriptors.

Location: localhost \ C: \ Program Files \ iWay55 \ bea

○	ibse
●	iwae
○	iwjcaivp

The following window opens.



Select targets for this Web application module

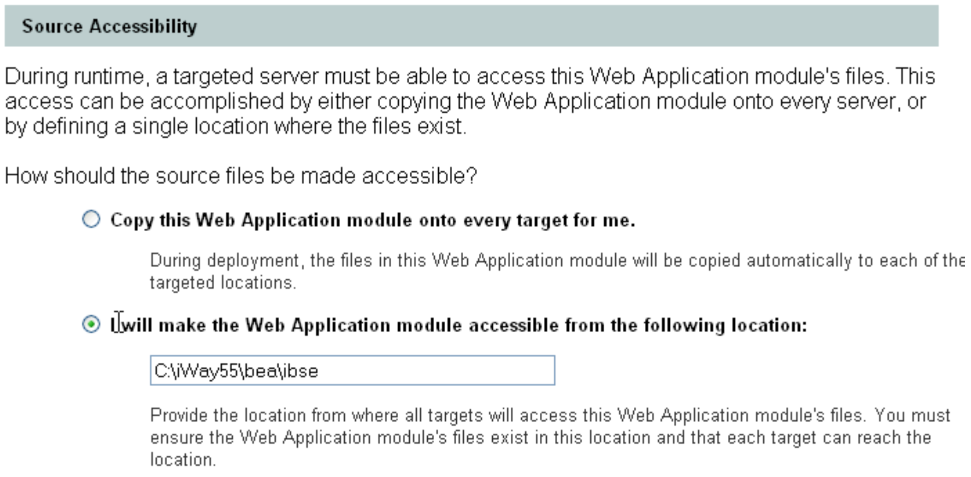
Select the servers and/or clusters on which you want to deploy your new Web Application module

Independent Servers
☐ AdminServer
☐ HTTPROUTER

Clusters
☑ MYCluster
● All servers in the cluster
○ Part of the cluster
☐ MS1
☐ MS2

6. Select the servers and/or clusters on which you want to deploy the application and click *Continue*.

   The following window opens.



Source Accessibility

During runtime, a targeted server must be able to access this Web Application module's files. This access can be accomplished by either copying the Web Application module onto every server, or by defining a single location where the files exist.

How should the source files be made accessible?

○ **Copy this Web Application module onto every target for me.**

During deployment, the files in this Web Application module will be copied automatically to each of the targeted locations.

● **I will make the Web Application module accessible from the following location:**

C:\iWay55\bea\ibse

Provide the location from where all targets will access this Web Application module's files. You must ensure the Web Application module's files exist in this location and that each target can reach the location.

7. Select the *I will make the Web Application module accessible from the following location* option button and provide the location from which all targets will access iBSE.

   It is recommended that you use a single instance of iBSE, rather than copying iBSE onto every target.

Note: iBSE must use a database repository (SQL or Oracle). Do not use a file repository. You can select this in the Repository Type drop-down list in the iBSE monitoring page. After configuring a database repository, you must restart all of the managed servers.

`http://`*`hostname`*`:`*`port`*`/ibse/IBSEConfig/`

where:

`hostname`

Is where your application server is running. Use the IP address or machine name in the URL; do not use localhost.

`port`

Is the port specific to each server, since you deploy iBSE to an entire cluster. For example, 8001, 8002, or any other port that is specified for each managed node.

8. Click *Deploy*.

Procedure How to Configure Ports and Channels in a Clustered Environment

You can use Swing Application Explorer deployed in BEA WebLogic WorkShop or Servlet Application Explorer to configure ports and channels in a clustered environment.

Note: Before using Servlet Application Explorer in a clustered environment, you must edit the web.xml file and specify the correct URL to your iBSE deployment. The default location on Windows is:

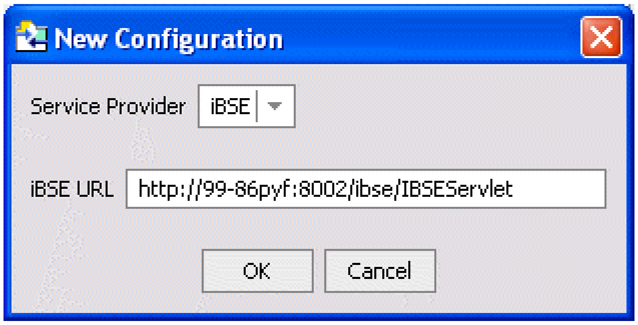
```
C:\Program Files\iWay55\bea\iwae\WEB-INF\web.xml
```

For more information on configuring the web.xml file for the Servlet Application Explorer, see the *BEA WebLogic ERP Adapter Installation and Configuration* documentation.

To configure ports and channels in a clustered environment:

1. Open Swing Application Explorer in BEA WebLogic Workshop.

2. Create a new connection to the iBSE instance. For information on creating a new configuration, see *Appendix A, Using Application Explorer in BEA WebLogic Workshop to Create XML Schemas and Web Services.*

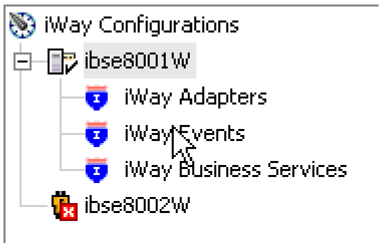


Service Provider: iBSE

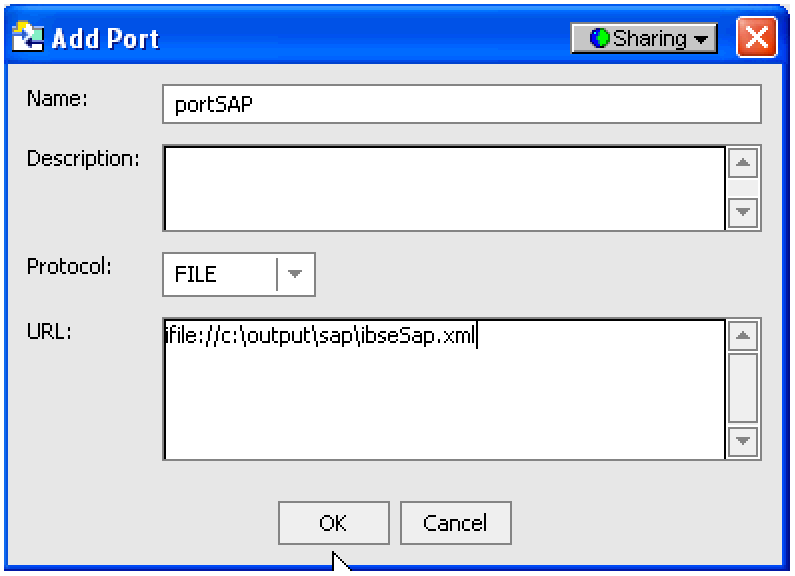
iBSE URL: http://99-86pyf:8002/ibse/IBSEServlet

Note: Use the IP address or machine name in the URL; do not use localhost.

3. Connect to the new configuration and select the Events node in the left pane of Application Explorer.



4. Select an adapter from the adapter list (in this example, SAP) and add a new port. For more information, see *Creating an Event Port* on page B-3.



Name: portSAP

Description:

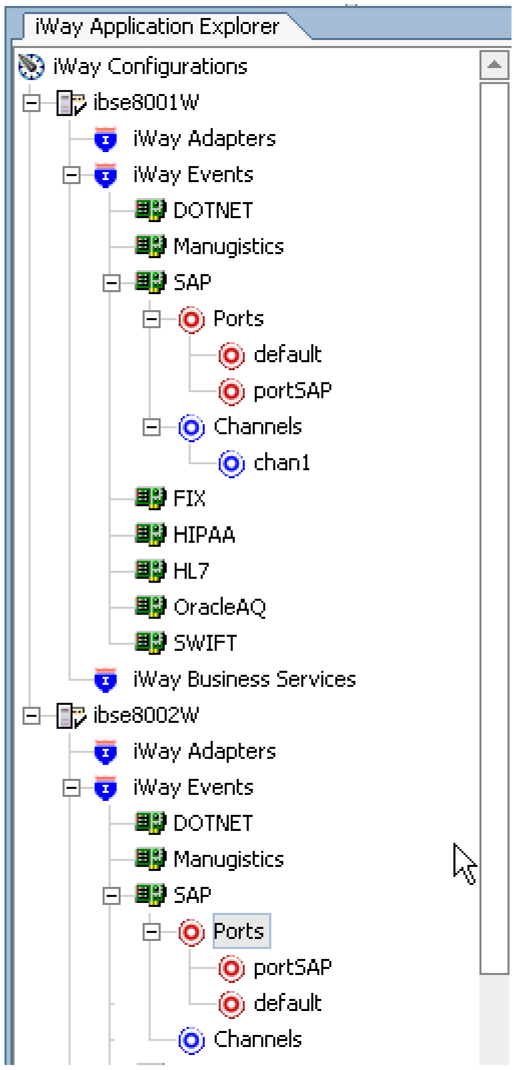
Protocol: FILE

URL: file://c:\output\sap\ibseSap.xml

5. Create a channel and add the port you created. For more information, see *Creating a Channel* on page B-18.
6. Click *Next* and enter the application server parameters.
7. Start the channel.
8. Create a new configuration and connect to the second iBSE instance.

   The connection to iBSE must be configured to each instance of the managed server.

The following graphic shows two configurations.



The following operations performed on one managed server will be replicated on all other managed servers:

- Create port and channel: Creates the channel and port under all available servers.
- Delete port and channel. Deletes the port and channel under all available servers.

The following operations must be performed on each server:

- Start channel. Starts the channel for the specific server.

- Stop channel. Stops the channel for the specific server.

Procedure How to Deploy JCA in a Clustered Environment

To deploy JCA in a clustered environment:

1. Using the BEA Configuration Wizard:

   a. Configure an administrative server to manage the managed servers.

   b. Add and configure as many managed servers as required.

   c. Add and configure an HTTP router. This does not have to be a part of WebLogic and can be an outside component.

   d. If you configure the HTTP router within WebLogic, start it by entering the following command:

```
StartManagedWebLogic HTTPROUTER http://localhost:7001
```

   where:

   `HTTPROUTER`

      Is the name of the server on which the HTTP router is running.

   `http://`*`localhost`*`:7001`

      Is the location of the admin console.

   e. Add the managed servers to your cluster/clusters.

For more information on configuring WebLogic Integration for deployment in a clustered environment, see *Deploying WebLogic Integration Solutions*.

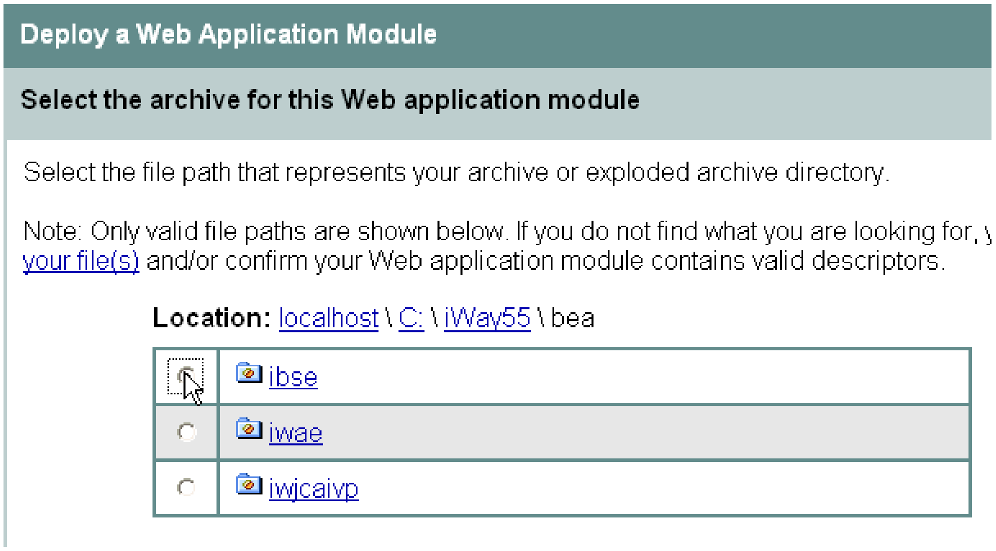
2. Start the WebLogic Server and open WebLogic Server Console.

3. Deploy iBSE to the cluster by selecting *Web Application Modules* from the Domain Configurations section, and clicking *Deploy a new Web Application Module*.

   A page appears for you to specify where the Web application is located.

   Note: You can deploy JCA to a cluster, but you can only point it to one directory, and to the machine on which it is installed.

4. To deploy iBSE, select the option button next to the ibse directory and then click *Target Module*.



Deploy a Web Application Module

Select the archive for this Web application module

Select the file path that represents your archive or exploded archive directory.

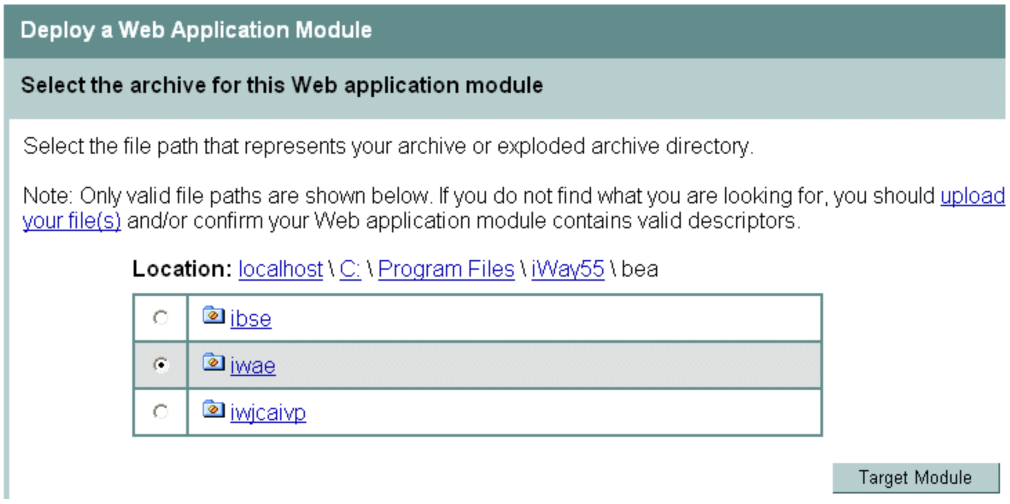
Note: Only valid file paths are shown below. If you do not find what you are looking for, y
your file(s) and/or confirm your Web application module contains valid descriptors.

Location: localhost \ C: \ iWay55 \ bea

◉	ibse
○	iwae
○	iwjcaivp

5. To deploy servlet Application Explorer, select the option button next to the iwae directory and then click *Target Module*.

   If you are using servlet Application Explorer, deploy it only on the admin server or one of the managed servers.



Deploy a Web Application Module

Select the archive for this Web application module

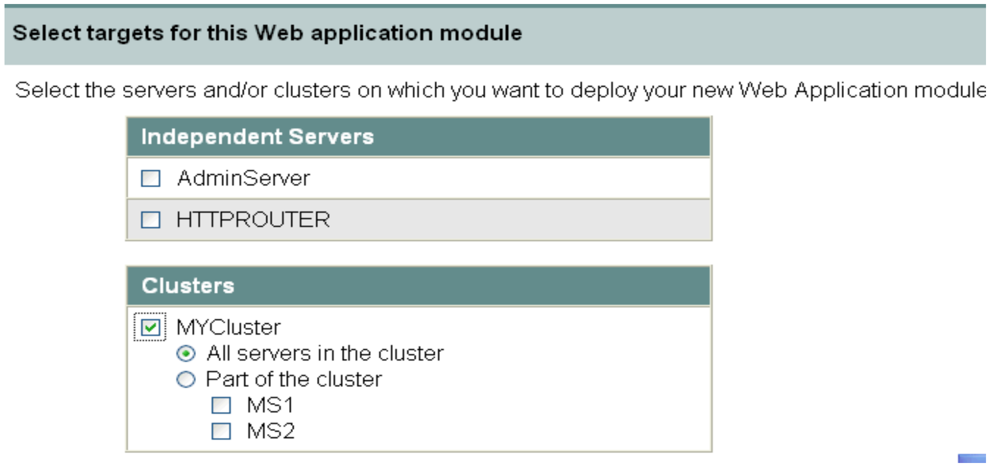
Select the file path that represents your archive or exploded archive directory.

Note: Only valid file paths are shown below. If you do not find what you are looking for, you should upload your file(s) and/or confirm your Web application module contains valid descriptors.

Location: localhost \ C: \ Program Files \ iWay55 \ bea

○	ibse
◉	iwae
○	iwjcaivp

The following window opens.



Select targets for this Web application module

Select the servers and/or clusters on which you want to deploy your new Web Application module

Independent Servers

☐ AdminServer

☐ HTTPROUTER

Clusters

☑ MYCluster

● All servers in the cluster

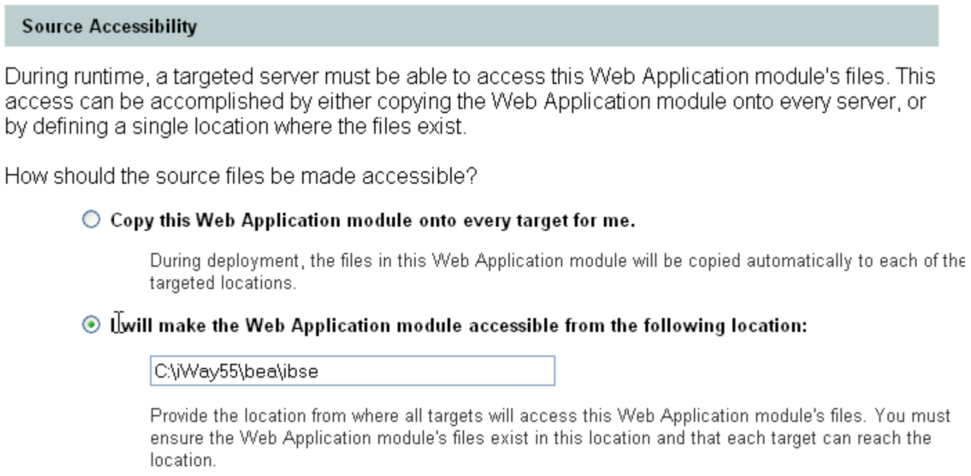
○ Part of the cluster

☐ MS1

☐ MS2

6. Select the servers and/or clusters on which you want to deploy the application and click *Continue*.

   The following window opens.



Source Accessibility

During runtime, a targeted server must be able to access this Web Application module's files. This access can be accomplished by either copying the Web Application module onto every server, or by defining a single location where the files exist.

How should the source files be made accessible?

○ **Copy this Web Application module onto every target for me.**

During deployment, the files in this Web Application module will be copied automatically to each of the targeted locations.

● **I will make the Web Application module accessible from the following location:**

C:\iWay55\bea\ibse

Provide the location from where all targets will access this Web Application module's files. You must ensure the Web Application module's files exist in this location and that each target can reach the location.

7. Select the *I will make the Web Application module accessible from the following location* option button and provide the location from which all targets will access iBSE.

   It is recommended that you use a single instance of iBSE, rather than copying iBSE onto every target.

Note: iBSE must use a database repository (SQL or Oracle). Do not use a file repository. You can select this in the Repository Type drop-down list in the iBSE monitoring page. After configuring a database repository, you must restart all of the managed servers.

`http://`*`hostname`*`:`*`port`*`/ibse/IBSEConfig/`

where:

`hostname`

Is where your application server is running. Use the IP address or machine name in the URL; do not use localhost.

`port`

Is the port specific to each server, since you deploy iBSE to an entire cluster. For example, 8001, 8002, or any other port that is specified for each managed node.

8. Click *Deploy*.

APPENDIX C

Using WebLogic Workshop to Access Web Services

Topics:

This section describes how to access Web services created for an SAP Business Application Programming Interface (BAPI) and an SAP Remote Function Call (RFC) using WebLogic Workshop.

Using WebLogic Workshop to Access an SAP BAPI or an SAP RFC

WebLogic Workshop provides a framework for building Web services. The Web services that you build with WebLogic Workshop are enterprise-class services, and WebLogic Workshop provides simple controls for connecting to your enterprise resources.

At the same time, WebLogic Workshop simplifies the process of creating Web services by insulating developers from the low-level implementation details that have traditionally made Web service development the domain of sophisticated J2EE developers. With WebLogic Workshop, you can build powerful Web services whether you are an application developer or a J2EE expert.

Accessing an SAP BAPI

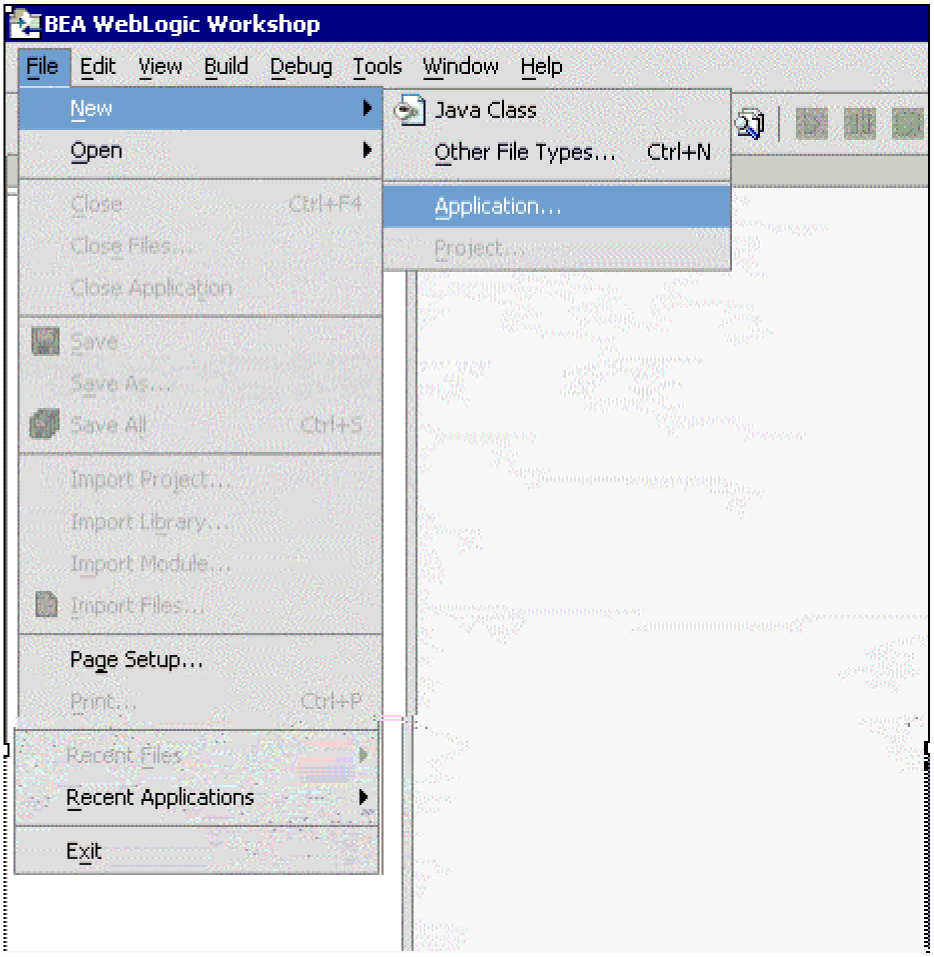
The following procedures assumes you already created and tested a Web service using Application Explorer. It also assumes you created the WSDL used to access the service. For more information on creating Web services, see Chapter 4, *Creating and Publishing Integration Business Services*.

Procedure How to Access an SAP BAPI

To access an SAP Business Application Programming Interface (BAPI):

1. From the Start menu, choose *Programs*, *WebLogic Platform 8.1*, and then *WebLogic Workshop* 8.1.

BEA WebLogic Workshop opens.



2. Create a new application.

   a. From the File menu, select *New* and then, *Application*.

   b. In the upper-left pane, select all and then, select *Empty Application*.

   c. In the directory field, type *C:\IWAYSRV*.

   d. Click *Create*.

3. In the Application tab, right-click the *IWAYSRV* folder and select *New Project*.

4. In the upper-left pane, select all and then, select *Web Project*.

5. In the name field, type *BAPIProj* and click *Create*.

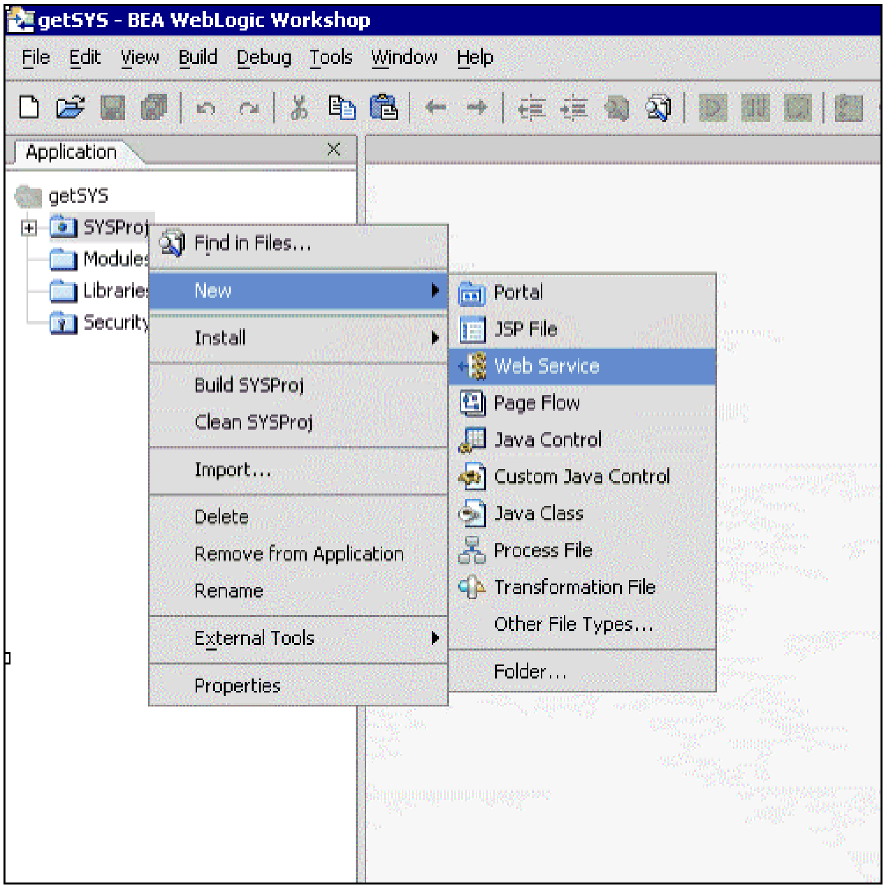
Calling a New Web Service for a BAPI

The code for a Web service is contained within a JWS (Java™ Web Service) file. A JWS file is a JAVA file in that it contains code for a Java class. However, because a file with a JWS extension contains the implementation code intended specifically for a Web service class, the extension gives it special meaning in the context of the WebLogic Server.

After you access the Business Application Programming Interface (BAPI), the New Web Service dialog box opens, and you can continue and call a Web service.

Procedure How to Call a New Web Service

To call a new Web service:

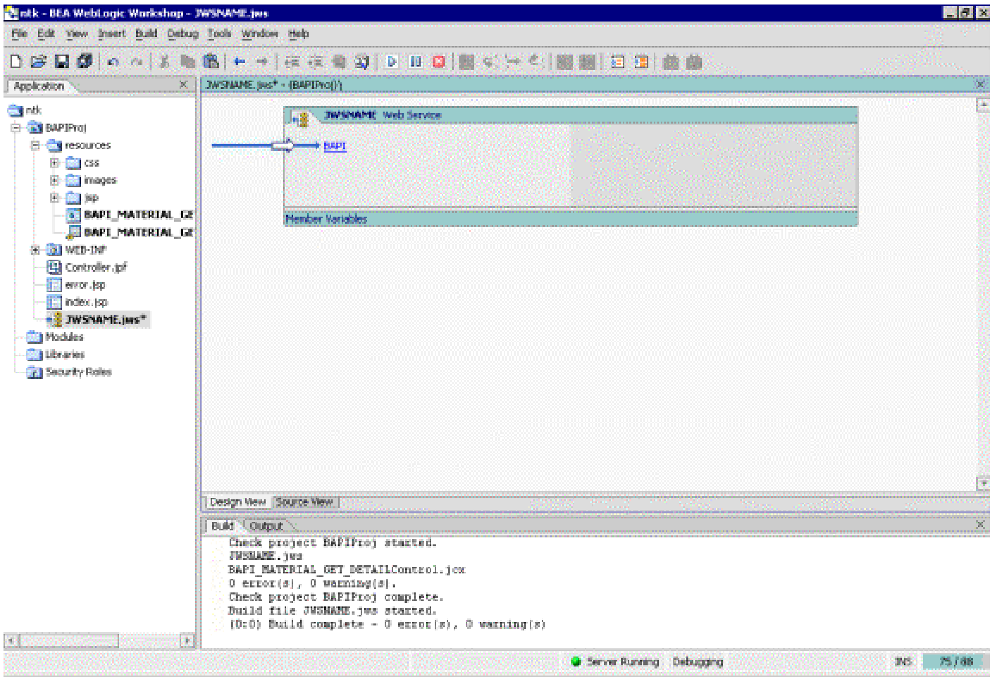


1. In the Application tab, right-click the *BAPIProj* folder and select *New*.
2. Select *Web Service*.

3. In the upper-left pane, select all and then, select *Web Service* in the right pane.
   - a. In the name field, type *JWSNAME.jws*.
   - b. Click *Create.*

     The design view window opens.

     Web services expose their functionality through methods that clients invoke when they want to request something from the Web service. In this case, clients invoke a method to call the BAPI_MATERIAL_GET_DETAIL Control that is exposed later in this procedure.
4. If it is not selected already, click the *Design View* tab.
   - a. From the Insert menu, select *Method.*
   - b. In the space provided, replace *method1* with *BAPI*, and press *Enter.*

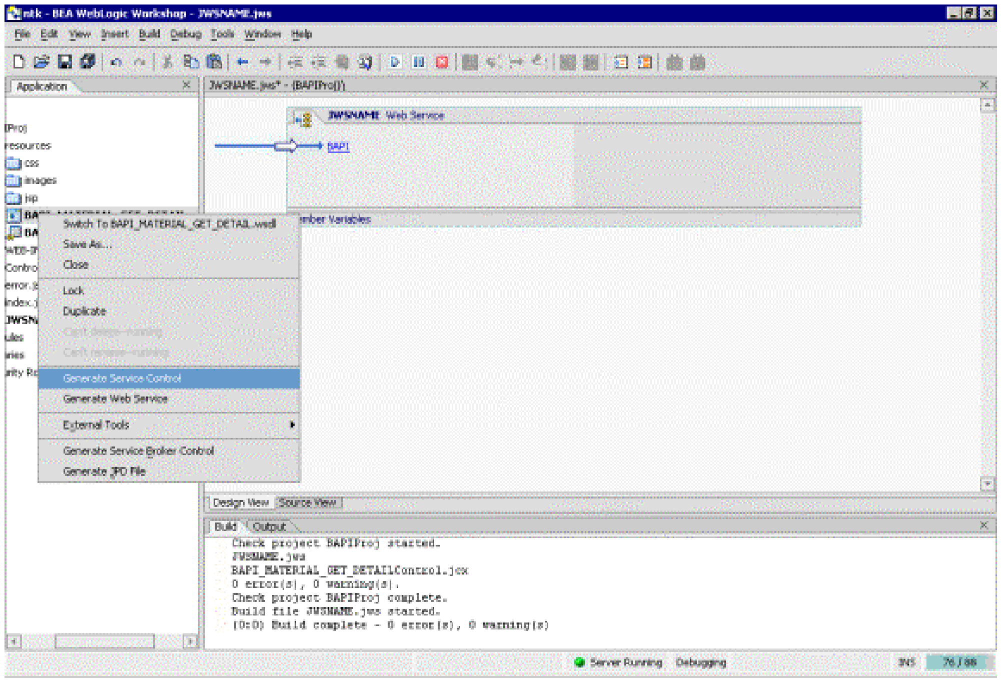


```
Check project BAPIProj started.
JWSNAME.jws
BAPI_MATERIAL_GET_DETAILControl.jcx
0 error(s), 0 warning(s).
Check project BAPIProj complete.
Build file JWSNAME.jws started.
(0:0) Build complete - 0 error(s), 0 warning(s)
```

5. Right-click the *resources* sub-folder project and select *Import.*
6. Import *BAPI_MATERIAL_GET_DETAIL.WSDL* .

   For more information on creating a WSDL file, see Chapter 4, *Creating and Publishing Integration Business Services*.

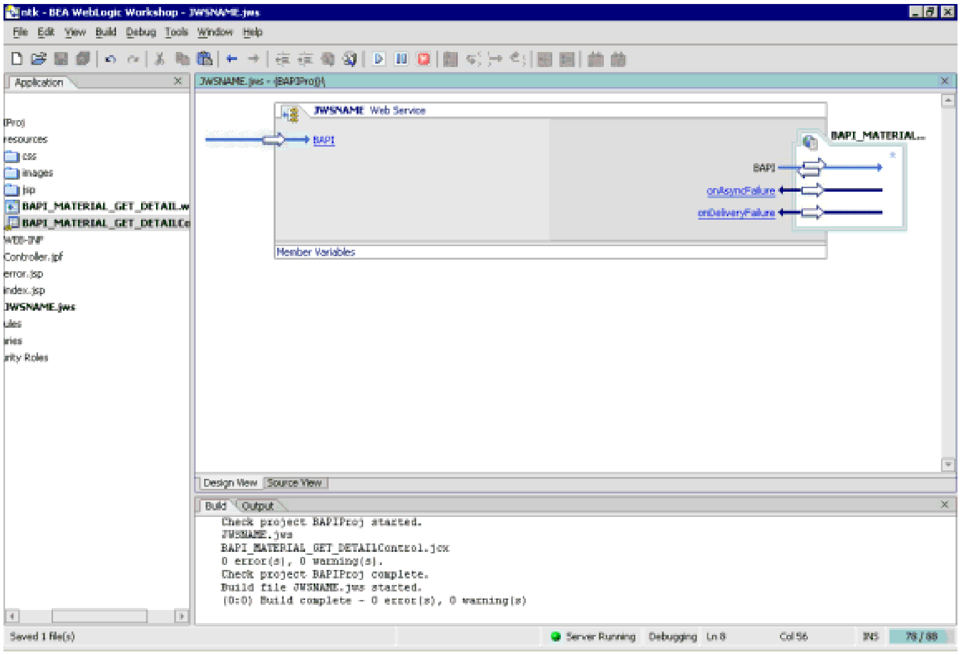
The following opens.



```
Check project BAPIProj started.
JWSNAME.jws
BAPI_MATERIAL_GET_DETAILControl.jcx
0 error(s), 0 warning(s).
Check project BAPIProj complete.
Build file JWSNAME.jws started.
(0:0) Build complete - 0 error(s), 0 warning(s)
```

7. To generate a Java Control file, right-click the *BAPI_MATERIAL_GET_DETAIL.wsdl* file and select *Generate Service Control*.

8. Drag the *BAPI_MATERIAL_GET_DETAIL.jcx* file onto the JWSNAME Web service as follows:



```
Check project BAPIProj started.
JWSNAME.jws
BAPI_MATERIAL_GET_DETAILControl.jcx
0 error(s), 0 warning(s).
Check project BAPIProj complete.
Build file JWSNAME.jws started.
(0:0) Build complete - 0 error(s), 0 warning(s)
```

9. Click the *Source View* tab to modify the source code and call the *BAPI_MATERIAL_GET_DETAIL* Web service.

10. Add the following code to the source view:

```
 public void
BAPI(BAPI_MATERIAL_GET_DETAILControl.BAPI_MATERIAL_GET_DETAIL input)
  {
    BAPI_MATERIAL_GET_DETAILControl.BAPI(input)
```

11. To save your current work, press *Control* + *S*.

The resulting Java code looks similar to the following:

```
import resources.BAPI_MATERIAL_GET_DETAILControl;

public class JWSNAME implements com.bea.jws.WebService
{
 /**

     * @common:control
     */
    private resources.BAPI_MATERIAL_GET_DETAILControl
BAPI_MATERIAL_GET_DETAILControl;




    static final long serialVersionUID = 1L;

    /**
     * @common:operation
     */
    public void
BAPI(BAPI_MATERIAL_GET_DETAILControl.BAPI_MATERIAL_GET_DETAIL input )
  {
    BAPI_MATERIAL_GET_DETAILControl.BAPI(input);
  }
}
```

Accessing an SAP RFC

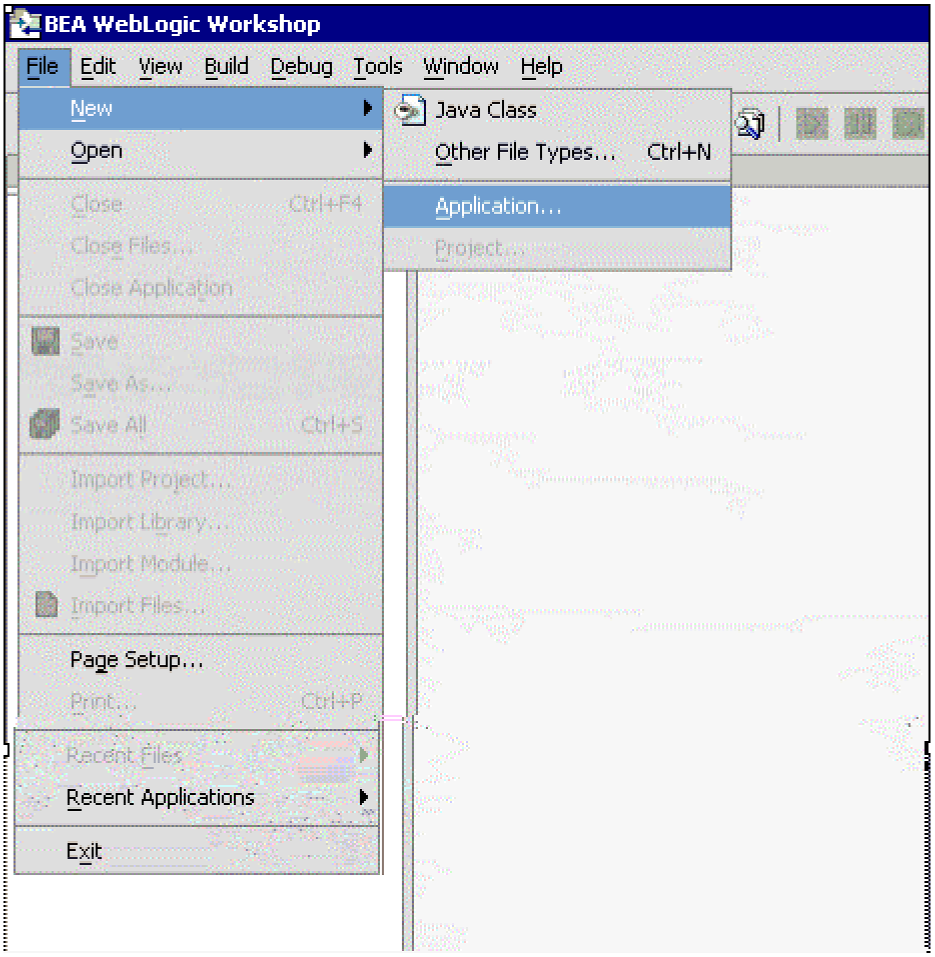
The following procedure describes how to access an SAP Remote Function Call (RFC) and assumes you already created and tested a Web service using Application Explorer. It also assumes you created the WSDL used to access the service. For more information on creating Web services, see Chapter 4, *Creating and Publishing Integration Business Services*.

Procedure How to Access an SAP RFC

To access an SAP Remote Function Call (RFC):

1. From the Start menu, choose *Programs*, *WebLogic Platform 8.1*, *WebLogic Workshop*, and then *WebLogic Workshop*.

BEA WebLogic Workshop opens.



2. Create a new application.
   a. From the File menu, select *New* and then, *Application*.
   b. In the upper-left pane, select all and then, select *Empty Application*.
   c. In the directory field, type *C:\IWAYSRV*.
   d. Click *Create*.
3. In the Application tab, right-click the *IWAYSRV* folder and select *New Project*.
4. In the upper-left pane, select all and then, select *Web Project*.
5. In the name field, type *RFCProj* and click *Create*.

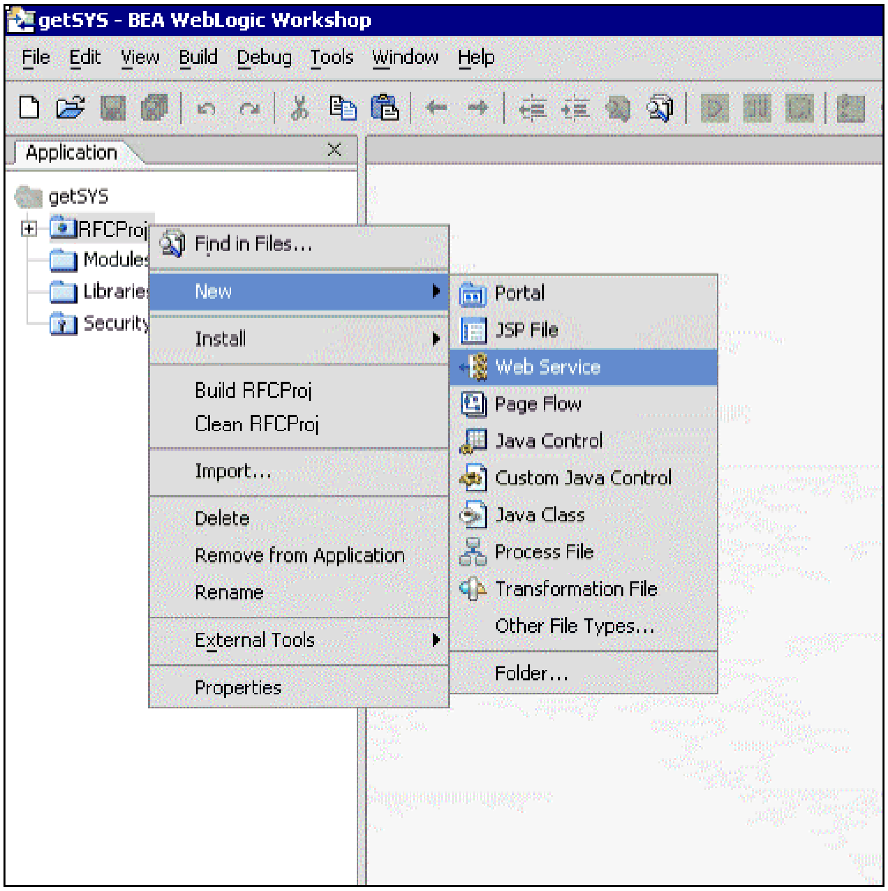
Calling a New Web Service for an RFC

The code for a Web service is contained within a JWS (Java Web Service) file. A JWS file is a JAVA file in that it contains code for a Java class. However, because a file with a JWS extension contains the implementation code intended specifically for a Web service class, the extension gives it special meaning in the context of the WebLogic Server.

After you access the Remote Function Call (RFC), the New Web Service dialog box opens, and you can continue and call a Web service.

Procedure How to Call a New Web Service

To call a new Web service:

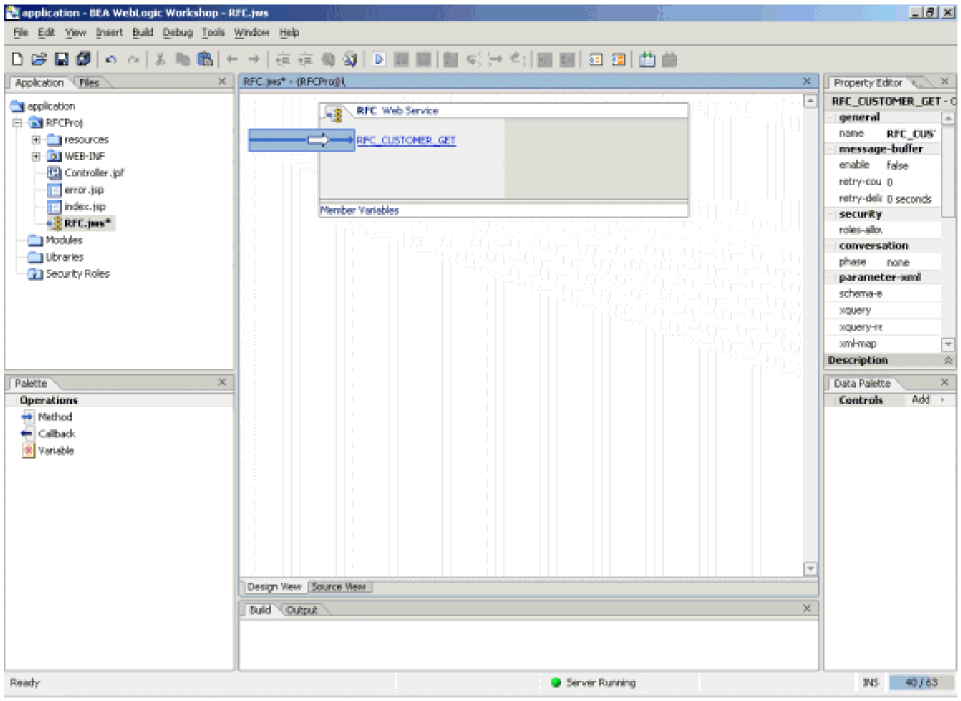


1. In the Application tab, right-click the *RFCProj* folder and select *New*.
2. Select *Web Service*.

3. In the upper-left pane, select all and then, select *Web Service* in the right pane.
   - **a.** In the name field, type *RFC.jws*.
   - **b.** Click *Create.*

     The design view window opens.

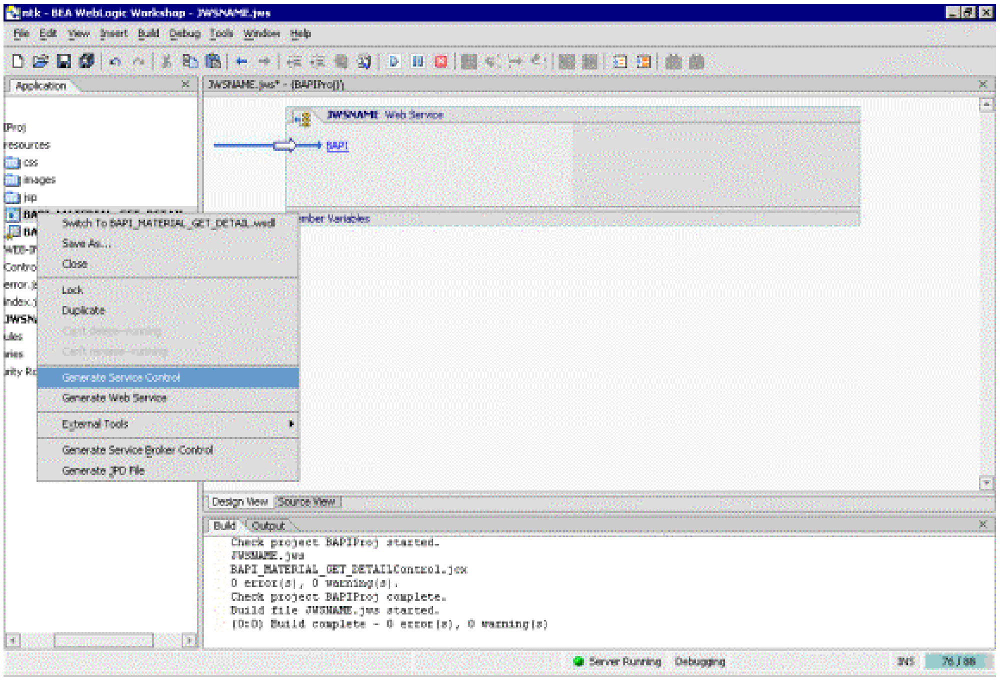
     Web services expose their functionality through methods that clients invoke when they want to request something from the Web service. In this case, clients invoke a method to call the RFC_CUSTOMER_GET Control that is exposed later in this procedure.
4. If it is not selected already, click the *Design View* tab.
   - **a.** From the Insert menu, select *Method*.
   - **b.** In the space provided, replace *method1* with *RFC_CUSTOMER_GET.*
5. Press *Enter.*



   - **a.** Right-click the *resources* sub-folder project and select *Import*.
   - **b.** Import the *RFC.WSDL*.

For more information on creating a WSDL file, see Chapter 4, *Creating and Publishing Integration Business Services*.

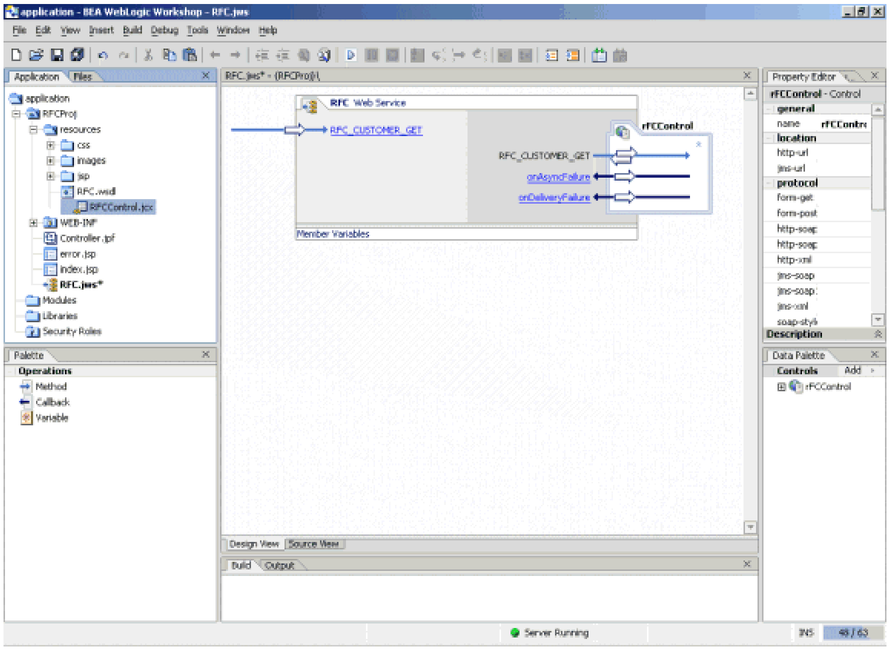
The following opens.



```
Check project BAPIProj started.
JWSNAME.jws
BAPI_MATERIAL_GET_DETAILControl.jcx
0 error(s), 0 warning(s).
Check project BAPIProj complete.
Build file JWSNAME.jws started.
(0:0) Build complete - 0 error(s), 0 warning(s)
```

6. To generate a Java Control file, right-click the *RFC_CUSTOMER_GET.wsdl* file and select *Generate Service Control*.

7. Drag the *RFC.jcx* file onto the JWSNAME Web service as follows:



8. Click the *Source View* tab to modify the source code and call the RFC Web service.

9. Add the following code to the source view:

```
public void RFC_CUSTOMER_GET(RFCControl.RFC_CUSTOMER_GET input)
  {
    RFCControl.RFC_CUSTOMER_GET(input)
```

10. To save your current work, press *Control* + *S*.

The resulting Java code looks similar to the following:

```
import resources.RFCControl;

public class RFC implements com.bea.jws.WebService
{
 /**


     * @common:control
     */
    private resources.RFCControl RFCControl;





    static final long serialVersionUID = 1L;

    /**
     * @common:operation
     */
    public void RFC_CUSTOMER_GET(RFCControl.RFC_CUSTOMER_GET input )
  {
    RFCControl.RFC_CUSTOMER_GET(input);
  }
}
```

Running the JWSNAME Web Service From WebLogic Workshop

When you create a new Web service tutorial application, you must ensure that WebLogic Server is running while you build your Web service.

Confirming WebLogic Server is Running

You can confirm whether WebLogic Server is running by looking at the status bar at the bottom of WebLogic Workshop. If WebLogic Server is running, a green ball appears. If WebLogic Server is not running, a red ball appears. If you see the red ball in the status bar, then start WebLogic Server, as described in the following procedure

Procedure How to Start WebLogic Server

To start WebLogic Server:

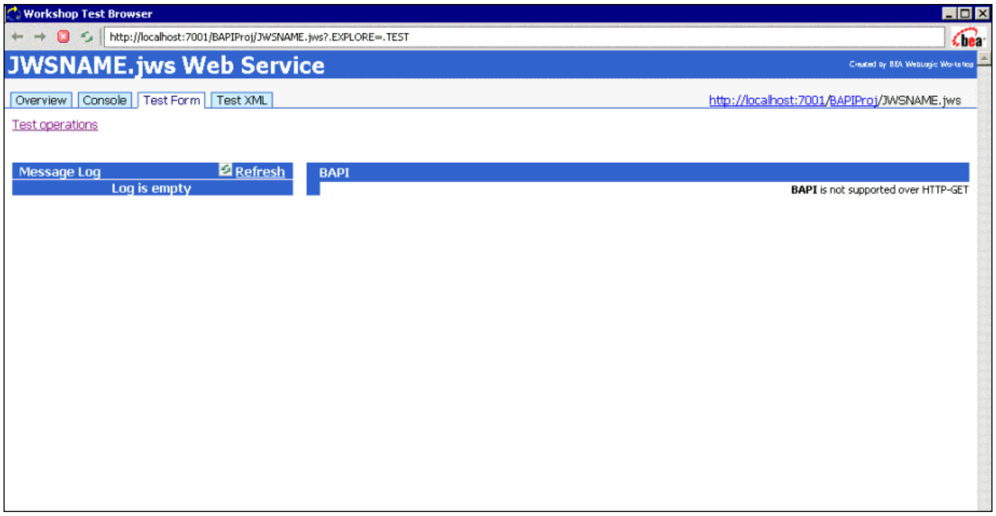
1. From the Tools menu, select *WebLogic Server* and then, *Start WebLogic Server.*
2. To deploy the application to WebLogic, select *Tools* and then, *Deploy Application*.
3. Click the *Start* button on the toolbar to start the application.

Running the JWSNAME Web Service for a BAPI

The following procedure describes how to run the JWSNAME for a Business Application Programming Interface (BAPI).

Procedure How to Run the JWSNAME Web Service for a BAPI

After you click the *Start* button on the tool bar to start the application, the following test window opens.



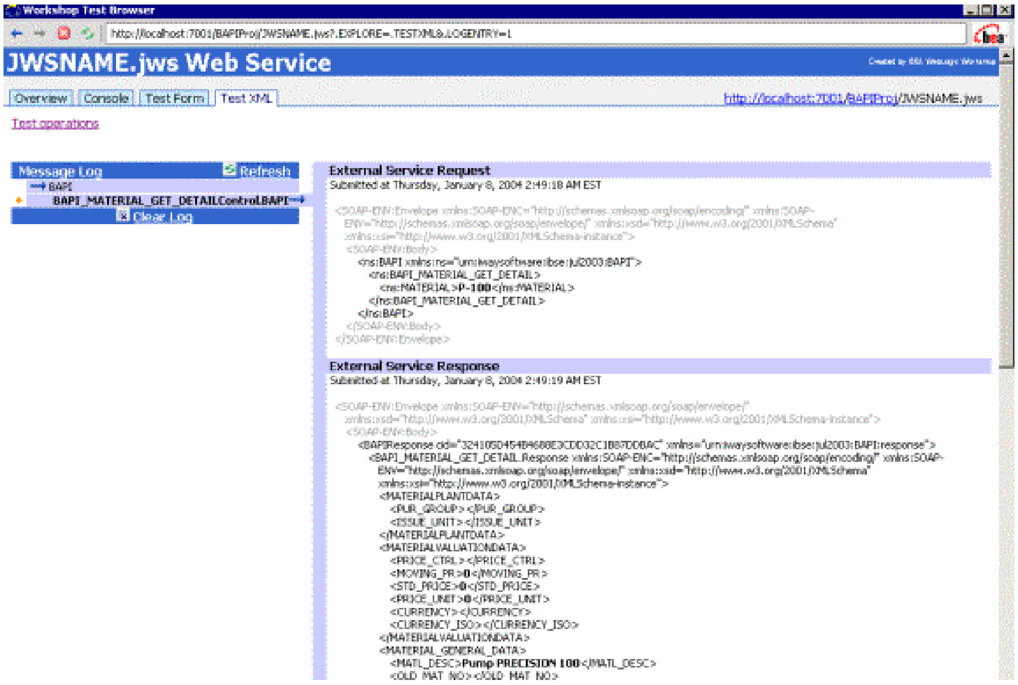
BAPI is not supported over HTTP-GET

1. Click the *Test XML* tab to enter and test the XML stream to be passed to the Web service.
2. Replace the string XML input with the following:

```
<?xml version="1.0"  encoding="UTF-8"?>
   <BAPI_MATERIAL_GET_DETAIL>
   <MATERIAL>P-100</MATERIAL>
<BAPI_MATERIAL_GET_DETAIL/>
```

3. Click the *BAPI* button to submit the request.

After the SOAP request is sent to the Integration Business Services Engine (iBSE), the following response is returned:


Workshop Test Browser
http://localhost:7001/BAPIProj/JWSNAME.jws?.EXPLORE=.TESTXML&.LOGENTRY=1
JWSNAME.jws Web Service
Overview
Console
Test Form
Test XML
http://localhost:7001/BAPIProj/JWSNAME.jws
Test operations
Message Log
Refresh
BAPI
BAPI_MATERIAL_GET_DETAILControlBAPI
Clear Log

External Service Request
Submitted at Thursday, January 8, 2004 2:49:18 AM EST

```
<SOAP-ENV:Envelope xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/" xmlns:SOAP-
  ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <SOAP-ENV:Body>
    <ns:BAPI xmlns:ns="urn:iwaysoftware:ibse:jul2003:BAPI">
      <ns:BAPI_MATERIAL_GET_DETAIL>
        <ns:MATERIAL>P-100</ns:MATERIAL>
      </ns:BAPI_MATERIAL_GET_DETAIL>
    </ns:BAPI>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

External Service Response
Submitted at Thursday, January 8, 2004 2:49:19 AM EST

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <SOAP-ENV:Body>
    <BAPIResponse cid="324105D454B4600E3CDD32C1B87DDBAC" xmlns="urn:iwaysoftware:ibse:jul2003:BAPI:response">
      <BAPI_MATERIAL_GET_DETAIL.Response xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/" xmlns:SOAP-
        ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
        <MATERIALPLANTDATA>
          <PUR_GROUP></PUR_GROUP>
          <ISSUE_UNIT></ISSUE_UNIT>
        </MATERIALPLANTDATA>
        <MATERIALVALUATIONDATA>
          <PRICE_CTRL></PRICE_CTRL>
          <MOVING_PR>0</MOVING_PR>
          <STD_PRICE>0</STD_PRICE>
          <PRICE_UNIT>0</PRICE_UNIT>
          <CURRENCY></CURRENCY>
          <CURRENCY_ISO></CURRENCY_ISO>
        </MATERIALVALUATIONDATA>
        <MATERIAL_GENERAL_DATA>
          <MATL_DESC>Pump PRECISION 100</MATL_DESC>
          <OLD_MAT_NO></OLD_MAT_NO>
```

The previous sample is a very simple example of calling a Web service. You may want to perform more complex operations in your workflow. For an example, see the following topic, *Calling Complex Operations in a Workflow for a BAPI*.

Calling Complex Operations in a Workflow for a BAPI

You may want to perform more complex operations in your workflow for a Business Application Programming Interface (BAPI). The following code represents sample Java code used to calculate the execution time of the Web service. You can do similar coding for benchmarking or other purposes.

```
import resources.BAPI_MATERIAL_GET_DETAILControl;

import java.io.*;
import java.lang.*;
import java.util.*;

public class JWSNAME implements com.bea.jws.WebService
{
    /**
     * @common:control
     */
    private resources.BAPI_MATERIAL_GET_DETAILControl
BAPI_MATERIAL_GET_DETAILControl;


    static final long serialVersionUID = 1L;

    /**
     * @common:operation
     */
    public void
BAPI(BAPI_MATERIAL_GET_DETAILControl.BAPI_MATERIAL_GET_DETAIL input)
throws Exception    {

    File outFile=new File("RESULTS.txt"); //creating an output file
    FileWriter out=new FileWriter(outFile); //creating a fileWriter for
the output file

    long diff=0; //used to store the execution time

   Calendar cal_start=Calendar.getInstance(TimeZone.getTimeZone("EST"));
//creating a start calendar
  System.out.println("<<<< start: "+ cal_start.getTimeInMillis());
//Display the start time of execution to the WEBLOGIC CONSOLE

         BAPI_MATERIAL_GET_DETAILControl.BAPI(input);

        Calendar
cal_end=Calendar.getInstance(TimeZone.getTimeZone("EST")); //create end
calendar
```

```
        System.out.println("<<<< end: "+ cal_end.getTimeInMillis());
Display the end time of execution to the WEBLOGIC CONSOLE
        diff=cal_end.getTimeInMillis()-cal_start.getTimeInMillis();
//Calculating the difference (execution time)
        System.out.println("<<<< EXECUTION time in Milliseconds:" +diff);
//Displaying the execution time to the WEBLOGIC Console

//writing to file
       out.write( "start time: "+ cal_start.getTimeInMillis()+"\n");
 out.write("end time: "+cal_end.getTimeInMillis()+"\n");
 out.write("execution time : "+diff+"\n");


    out.close(); //closing file



    }
}
```

The results of the execution are saved in a file as follows:

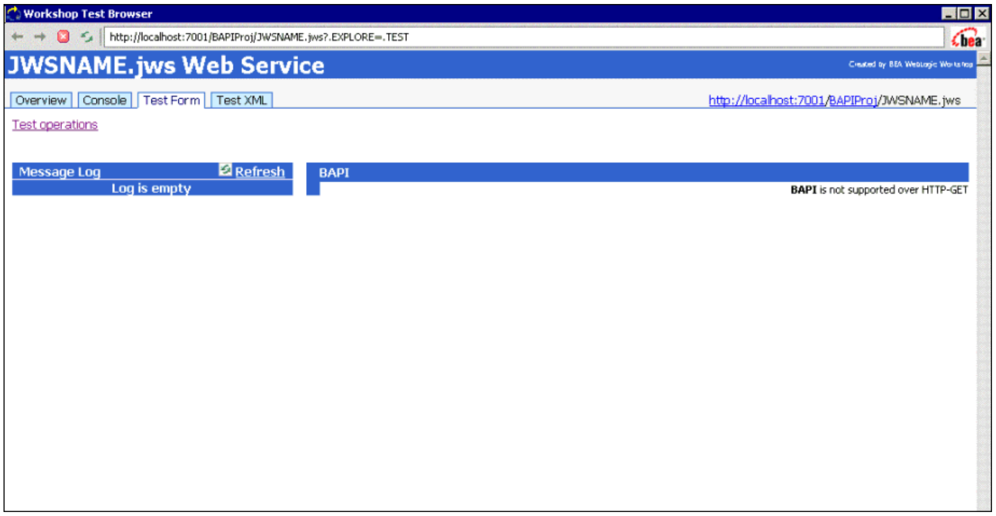
```
start time: 1073598362655
end time: 1073598362775
execution time : 120
```

Running the JWSNAME Web Service for an RFC

The following procedure describes how to run the JWSNAME for a Remote Function Call (RFC). You first must ensure that WebLogic Server is running. For more information on confirming that WebLogic Server is running, see *Confirming WebLogic Server is Running* and *How to Start WebLogic Server*.

Procedure How to Run the JWSNAME Web Service for an RFC

After you click the *Start* button on the tool bar to start the application, the following test window opens.


Workshop Test Browser
http://localhost:7001/BAPIProj/JWSNAME.jws?.EXPLORE=.TEST
JWSNAME.jws Web Service
Created by BEA WebLogic Workshop
Overview
Console
Test Form
Test XML
http://localhost:7001/BAPIProj/JWSNAME.jws
Test operations
Message Log
Refresh
Log is empty
BAPI

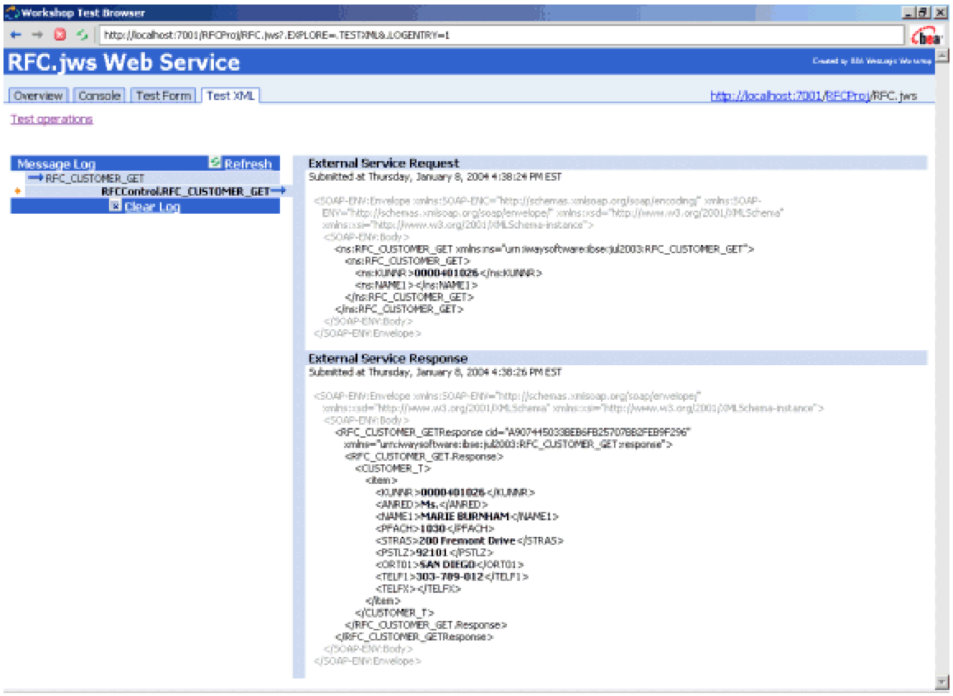
BAPI is not supported over HTTP-GET

1. Click the *Test XML* tab to enter and test the XML stream to be passed to the Web service.
2. Replace the string XML input with the following:

```
<RFC_CUSTOMER_GET xmlns="http://www.openuri.org/">
 <input>
   <KUNNR>0000401026</KUNNR>
    <NAME1></NAME1>
 </input>
</RFC_CUSTOMER_GET>
```

3. Click the *RFC_CUSTOMER_GET* button to submit the request.

After the SOAP request is sent to the Integration Business Services Engine (iBSE), the following response is returned:


Workshop Test Browser
http://localhost:7001/RFCProj/RFC.jws?.EXPLORE=.TESTXML&.LOGENTRY=1
RFC.jws Web Service
Overview
Console
Test Form
Test XML
http://localhost:7001/RFCProj/RFC.jws
Test operations
Message Log
Refresh
RFC_CUSTOMER_GET
RFCControl.RFC_CUSTOMER_GET
Clear Log

External Service Request
Submitted at Thursday, January 8, 2004 4:38:24 PM EST

```
<SOAP-ENV:Envelope xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/" xmlns:SOAP-
  ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <SOAP-ENV:Body>
    <ns:RFC_CUSTOMER_GET xmlns:ns="urn:iwaysoftware:ibse:jul2003:RFC_CUSTOMER_GET">
      <ns:RFC_CUSTOMER_GET>
        <ns:KUNNR>0000401026</ns:KUNNR>
        <ns:NAME1></ns:NAME1>
      </ns:RFC_CUSTOMER_GET>
    </ns:RFC_CUSTOMER_GET>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

External Service Response
Submitted at Thursday, January 8, 2004 4:38:26 PM EST

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <SOAP-ENV:Body>
    <RFC_CUSTOMER_GETResponse cid="A907445033BEB6FB25707B82FEB9F296"
      xmlns="urn:iwaysoftware:ibse:jul2003:RFC_CUSTOMER_GET:response">
      <RFC_CUSTOMER_GET.Response>
        <CUSTOMER_T>
          <item>
            <KUNNR>0000401026</KUNNR>
            <ANRED>Ms.</ANRED>
            <NAME1>MARIE BURNHAM</NAME1>
            <PFACH>1030</PFACH>
            <STRAS>200 Fremont Drive</STRAS>
            <PSTLZ>92101</PSTLZ>
            <ORT01>SAN DIEGO</ORT01>
            <TELF1>303-709-012</TELF1>
            <TELFX></TELFX>
          </item>
        </CUSTOMER_T>
      </RFC_CUSTOMER_GET.Response>
    </RFC_CUSTOMER_GETResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

The previous sample is a very simple example of calling a Web service.

You may want to perform more complex operations in your workflow.

Calling Complex Operations in a Workflow for an RFC

You may want to perform more complex operations in your workflow for a Remote Function Call (RFC). The following code represents sample Java code used to calculate the execution time of the Web service. You can do similar coding for benchmarking or other purposes.

```
import resources.RFCControl;

import java.io.*;
import java.lang.*;
import java.util.*;
```

```
public class RFC implements com.bea.jws.WebService
{
    /**
     * @common:control
     */
    private resources.RFCControl RFCControl;


    static final long serialVersionUID = 1L;

    /**
     * @common:operation
     */
    public void RFC_CUSTOMER_GET(RFCControl.RFC_CUSTOMER_GET input)
throws Exception    {

    File outFile=new File("RESULTS.txt"); //creating an output file
    FileWriter out=new FileWriter(outFile); //creating a fileWriter for
the output file

    long diff=0; //used to store the execution time

   Calendar cal_start=Calendar.getInstance(TimeZone.getTimeZone("EST"));
//creating a start calendar
  System.out.println("<<<< start: "+ cal_start.getTimeInMillis());
//Display the start time of execution to the WEBLOGIC CONSOLE

         RFCControl.RFC_CUSTOMER_GET(input);

        Calendar
cal_end=Calendar.getInstance(TimeZone.getTimeZone("EST")); //create end
calendar

        System.out.println("<<<< end: "+ cal_end.getTimeInMillis());
Display the end time of execution to the WEBLOGIC CONSOLE
        diff=cal_end.getTimeInMillis()-cal_start.getTimeInMillis();
//Calculating the difference (execution time)
        System.out.println("<<<< EXECUTION time in Milliseconds:" +diff);
//Displaying the execution time to the WEBLOGIC Console

//writing to file
       out.write( "start time: "+ cal_start.getTimeInMillis()+"\n");
 out.write("end time: "+cal_end.getTimeInMillis()+"\n");
 out.write("execution time : "+diff+"\n");


    out.close(); //closing file
    }
}
```

The results of the execution are saved in a file as follows:

```
start time: 1073598362650
end time: 1073598362775
execution time : 125
```

APPENDIX D

Sample Files and Coding Techniques

Topics:

This section provides sample request and response documents sent between SAP and the BEA WebLogic Adapter for SAP. It also includes a sample RFC module and a sample wrapper module to call functions on remote destinations.

Sample RFC Request Document

The following sample code shows a sample RFC request document.

```
<?xml version="1.0" ?>
<doc:RFC_WALK_THRU_TEST xmlns:doc="urn:sapcom:document:sap:business:rfc">
   <TEST_IN>
      <RFCFLOAT>0.0</RFCFLOAT>
      <RFCCHAR1></RFCCHAR1>
      <RFCINT2>0</RFCINT2>
      <RFCINT1>0</RFCINT1>
      <RFCCHAR4></RFCCHAR4>
      <RFCINT4>10</RFCINT4>
      <RFCHEX3>000000</RFCHEX3>
      <RFCCHAR2></RFCCHAR2>
      <RFCTIME>10:09:32</RFCTIME>
      <RFCDATE>2001-09-05</RFCDATE>
      <RFCDATA1>Hello World</RFCDATA1>
      <RFCDATA2></RFCDATA2>
   </TEST_IN>
   <DESTINATIONS>
   </DESTINATIONS>
   <LOG>
   </LOG>
</doc:RFC_WALK_THRU_TEST>
```

Sample RFC Response Document

The following sample code shows a sample RFC response document.

```
<?xml version="1.0" ?>
<doc:RFC_WALK_THRU_TEST.Response
 xmlns:doc="urn:sapcom:document:sap:business:rfc">
   <TEST_OUT>
      <RFCFLOAT>0.0</RFCFLOAT>
      <RFCCHAR1></RFCCHAR1>
      <RFCINT2>0</RFCINT2>
      <RFCINT1>0</RFCINT1>
      <RFCCHAR4></RFCCHAR4>
      <RFCINT4>10</RFCINT4>
      <RFCHEX3>000000</RFCHEX3>
      <RFCCHAR2></RFCCHAR2>
      <RFCTIME>10:09:32</RFCTIME>
      <RFCDATE>2001-09-05</RFCDATE>
      <RFCDATA1>Hello World</RFCDATA1>
      <RFCDATA2></RFCDATA2>
   </TEST_OUT>
   <DESTINATIONS>
   </DESTINATIONS>
   <LOG>
   </LOG>
</doc:RFC_WALK_THRU_TEST.Response>
```

Sample IDoc XML for Message Type DEBMAS

The following sample code shows a sample IDoc XML document for the DEBMAS message type.

```
<?xml version="1.0" ?>
<DEBMAS01>
   <IDOC BEGIN="1">
      <EDI_DC40 SEGMENT="1">
         <TABNAM>EDI_DC40</TABNAM>
         <MANDT>800</MANDT>
<DOCNUM>0000000000236015</DOCNUM>
         <DOCREL>46C</DOCREL>
         <STATUS>30</STATUS>
         <DIRECT>1</DIRECT>
         <OUTMOD>2</OUTMOD>
         <EXPRSS></EXPRSS>
         <TEST></TEST>

         <IDOCTYP>DEBMAS01</IDOCTYP>
         <CIMTYP></CIMTYP>
         <MESTYP>DEBMAS</MESTYP>
         <MESCOD></MESCOD>
         <MESFCT></MESFCT>
         <STD></STD>

         <STDVRS></STDVRS>
         <STDMES></STDMES>
         <SNDPOR>SAPI46</SNDPOR>
         <SNDPRT>LS</SNDPRT>
         <SNDPFC></SNDPFC>
         <SNDPRN>I46_CLI800</SNDPRN>
         <SNDSAD></SNDSAD>
         <SNDLAD></SNDLAD>

         <RCVPOR>A000000018</RCVPOR>
         <RCVPRT>LS</RCVPRT>
         <RCVPFC></RCVPFC>
         <RCVPRN>SAMP</RCVPRN>
         <RCVSAD></RCVSAD>
         <RCVLAD></RCVLAD>
         <CREDAT>2001-09-04</CREDAT>
         <CRETIM>16:44:52</CRETIM>
         <REFINT></REFINT>
         <REFGRP></REFGRP>
         <REFMES></REFMES>
         <ARCKEY></ARCKEY>
         <SERIAL>20010904164452</SERIAL>
     </EDI_DC40>
```

```
<E1KNA1M SEGMENT="1">
   <MSGFN>005</MSGFN>
   <KUNNR>0000000001</KUNNR>
   <ANRED></ANRED>
   <AUFSD></AUFSD>
   <BAHNE></BAHNE>
   <BAHNS></BAHNS>
   <BBBNR>0000000</BBBNR>
   <BBSNR>00000</BBSNR>

   <BEGRU></BEGRU>
   <BRSCH></BRSCH>
   <BUBKZ>0</BUBKZ>
   <DATLT></DATLT>
   <FAKSD></FAKSD>
   <FISKN></FISKN>
   <KNRZA></KNRZA>
   <KONZS></KONZS>

   <KTOKD>0001</KTOKD>
   <KUKLA></KUKLA>

   <LAND1>US</LAND1>
   <LIFNR></LIFNR>
   <LIFSD></LIFSD>
   <LOCCO></LOCCO>
   <LOEVM></LOEVM>
   <NAME1>Apple Corp</NAME1>
   <NAME2></NAME2>
   <NAME3></NAME3>
   <NAME4></NAME4>

   <NIELS></NIELS>
   <ORT01>Floral Park</ORT01>
   <ORT02></ORT02>
   <PFACH></PFACH>
   <PSTL2></PSTL2>
   <PSTLZ>10010</PSTLZ>
   <REGIO>NY</REGIO>
   <COUNC></COUNC>
   <CITYC></CITYC>
   <RPMKR></RPMKR>
   <SORTL>APPLE</SORTL>
   <SPERR></SPERR>
   <SPRAS>E</SPRAS>
   <STCD1></STCD1>
   <STCD2></STCD2>
   <STKZA></STKZA>
   <STKZU></STKZU>
   <STRAS>123 Main street</STRAS>
```

```
<TELBX></TELBX>
<TELF1></TELF1>
<TELF2></TELF2>
<TELFX></TELFX>
<TELTX></TELTX>
<TELX1></TELX1>
<LZONE>0000000001</LZONE>

<XZEMP></XZEMP>
<VBUND></VBUND>
<STCEG></STCEG>
<GFORM></GFORM>
<BRAN1></BRAN1>
<BRAN2></BRAN2>
<BRAN3></BRAN3>
<BRAN4></BRAN4>
<BRAN5></BRAN5>
<UMJAH>0000</UMJAH>

<UWAER></UWAER>
<JMZAH>000000</JMZAH>
<JMJAH>0000</JMJAH>
<KATR1></KATR1>
<KATR2></KATR2>
<KATR3></KATR3>
<KATR4></KATR4>
<KATR5></KATR5>
<KATR6></KATR6>
<KATR7></KATR7>
<KATR8></KATR8>
<KATR9></KATR9>
<KATR10></KATR10>
<STKZN></STKZN>
```

```
 <UMSA1>0</UMSA1>
 <TXJCD></TXJCD>
 <PERIV></PERIV>
 <KTOCD></KTOCD>
 <PFORT></PFORT>
 <DTAMS></DTAMS>
 <DTAWS></DTAWS>
 <HZUOR>00</HZUOR>
 <CIVVE>X</CIVVE>
 <MILVE></MILVE>
 <SPRAS_ISO>EN</SPRAS_ISO>
 <FITYP></FITYP>
 <STCDT></STCDT>
 <STCD3></STCD3>
 <STCD4></STCD4>
 <XICMS></XICMS>
 <CFOPC></CFOPC>

 <TXLW1></TXLW1>
 <TXLW2></TXLW2>
 <CCC01></CCC01>
 <CCC02></CCC02>
 <CCC03></CCC03>
 <CCC04></CCC04>
 <CASSD></CASSD>
 <KDKG1></KDKG1>
 <KDKG2></KDKG2>
 <KDKG3></KDKG3>
 <KDKG4></KDKG4>
 <KDKG5></KDKG5>
 <NODEL></NODEL>
 <XSUB2></XSUB2>
<WERKS></WERKS>
```

```
<E1KNVVM SEGMENT="1">
   <MSGFN>005</MSGFN>
   <VKORG>0001</VKORG>
   <VTWEG>01</VTWEG>
   <SPART>01</SPART>
  <BEGRU></BEGRU>
  <LOEVM></LOEVM>
  <VERSG></VERSG>
  <AUFSD></AUFSD>
  <KALKS>1</KALKS>
  <KDGRP></KDGRP>
  <BZIRK></BZIRK>
  <KONDA></KONDA>
  <PLTYP></PLTYP>
  <AWAHR>100</AWAHR>
  <INCO1></INCO1>
  <INCO2></INCO2>
  <LIFSD></LIFSD>
  <AUTLF></AUTLF>
  <ANTLF>9</ANTLF>
  <KZTLF></KZTLF>
  <KZAZU>X</KZAZU>
  <CHSPL></CHSPL>
  <LPRIO>00</LPRIO>
  <EIKTO></EIKTO>
  <VSBED>01</VSBED>
  <FAKSD></FAKSD>
  <MRNKZ></MRNKZ>
  <PERFK></PERFK>
  <PERRL></PERRL>
  <WAERS>EUR</WAERS>
  <KTGRD></KTGRD>

   <ZTERM></ZTERM>
   <VWERK></VWERK>
   <VKGRP></VKGRP>
   <VKBUR></VKBUR>
   <VSORT></VSORT>
   <KVGR1></KVGR1>
   <KVGR2></KVGR2>
   <KVGR3></KVGR3>
   <KVGR4></KVGR4>
   <KVGR5></KVGR5>
```

```
<BOKRE></BOKRE>
<KURST></KURST>
<PRFRE></PRFRE>
<KLABC></KLABC>
<KABSS></KABSS>
<KKBER></KKBER>
<CASSD></CASSD>
<RDOFF></RDOFF>
<AGREL></AGREL>
<MEGRU></MEGRU>
<UEBTO>0.0</UEBTO>
<UNTTO>0.0</UNTTO>
<UEBTK></UEBTK>
<PVKSM></PVKSM>
<PODKZ></PODKZ>
<PODTG>         0</PODTG>

<E1KNVPM SEGMENT="1">
    <MSGFN>005</MSGFN>
    <PARVW>AG</PARVW>
    <KUNN2>0000000001</KUNN2>
    <DEFPA></DEFPA>
    <KNREF></KNREF>
    <PARZA>000</PARZA>
</E1KNVPM>

<E1KNVPM SEGMENT="1">
    <MSGFN>005</MSGFN>
    <PARVW>RE</PARVW>
    <KUNN2>0000000001</KUNN2>
    <DEFPA></DEFPA>
    <KNREF></KNREF>
    <PARZA>000</PARZA>
</E1KNVPM>

<E1KNVPM SEGMENT="1">
    <MSGFN>005</MSGFN>
    <PARVW>RG</PARVW>
    <KUNN2>0000000001</KUNN2>
    <DEFPA></DEFPA>
    <KNREF></KNREF>
    <PARZA>000</PARZA>
</E1KNVPM>
```

```
        <E1KNVPM SEGMENT="1">
            <MSGFN>005</MSGFN>
            <PARVW>WE</PARVW>
            <KUNN2>0000000001</KUNN2>
            <DEFPA></DEFPA>
            <KNREF></KNREF>
            <PARZA>000</PARZA>
        </E1KNVPM>

        <E1KNVIM SEGMENT="1">
             <MSGFN>005</MSGFN>
             <ALAND>DE</ALAND>
             <TATYP>MWST</TATYP>
             <TAXKD>0</TAXKD>
        </E1KNVIM>

     </E1KNVVM>
   </E1KNA1M>
  </IDOC>
</DEBMAS01
```

Collected IDocs

When using collected IDocs on any platform during inbound processing (service mode), if the DOCNUM field does not have a unique document number for each IDoc, the system creates an IDoc for each header record in the collected IDoc file and duplicates the data for each IDoc.

Make sure the DOCNUM field is included in the EDI_DC40 structure and that each IDoc has a unique sequence number within the collected IDoc file.

Sample RFC Module

After you have configured the SAP event adapter and the RFC destination, you can write ABAP code to execute calls at your new destination (the event adapter).

The following sample code uses a user-defined RFC module named Z_EVENT_DISPATCH.

```
FUNCTION Z_01_EVENT_DISPATCH.
CALL FUNCTION 'Z_EVENT_DISPATCH'
  DESTINATION 'IWAYDEST'
  EXPORTING
    EVENT = EVENT
    RECTYPE = RECTYPE
    OBJTYPE = OBJTYPE
    OBJKEY = OBJKEY
  TABLES
    EVENT_CONTAINER = EVENT_CONTAINER.
ENDFUNCTION.
```

Sample Wrapper Module to Call Functions on Remote Destinations (Service)

This topic describes how to invoke a service that employs SAP remote data. For example, you can use this technique to write a function using C on a UNIX server that queries an Informix database and returns the response to SAP.

The ABAP command, CALL FUNCTION, takes as an argument, DESTINATION. Using RFC (Remote Function Call) destinations, programs can be executed on external systems, and the results can be returned into SAP function module programs. For more information on this functionality, see your SAP documentation, which is available at the following URL:

`http://help.sap.com`

Since DESTINATION is not part of an individual BAPI (Business Application Programming Interface) or Remote Function Module (RFM), but a parameter of the SAP function mechanism, you require a *wrapper module* to invoke it as a service. In addition, you must invoke the wrapper module in place of the original function.

The wrapper module is written using SAP's ABAP/4 programming language and contains the same input and output parameters as the original function. You can obtain all of the parameters of a remote function in the function editor by selecting Edit and then, Pattern and entering the function name.

The destination inside the wrapper module must be a valid SAP RFC destination with an RFC Server program running on the remote host. For more information, see SAP's RFC Programming manual, which is available at the following URL:

`http://service.sap.com`

The RFC Server program must return the data to SAP in a format that follows the exact structure of the Remote Function interface, or an abnormal ending occurs in SAP.

The following is an example of a wrapper module for the SAP test function named RFC_CUSTOMER_GET.

```
FUNCTION Z_CALL_EXTERNAL.
*"------------------------------------------------------------------------
*"*"Local interface:
*"   IMPORTING
*"      VALUE(MYKUNNR) LIKE  KNA1-KUNNR DEFAULT SPACE
*"      VALUE(MYNAME1) LIKE  KNA1-NAME1 DEFAULT SPACE
*"   EXPORTING
*"      VALUE(ERRORCODE) LIKE  SY-SUBRC
*"   TABLES
*"       MYCUSTOMER_T STRUCTURE  BRFCKNA1
*"------------------------------------------------------------------------
ERRORCODE = 0.
CALL FUNCTION 'RFC_CUSTOMER_GET'
 DESTINATION 'JRDEST'
  EXPORTING
    KUNNR                     = MYKUNNR
    NAME1                     = MYNAME1
    TABLES
    CUSTOMER_T                = MYCUSTOMER_T
EXCEPTIONS
      COMMUNICATION_FAILURE = -1
      SYSTEM_FAILURE        = -2
      NOTHING_SPECIFIED     = -3
      NO_RECORD_FOUND       = -4
      OTHERS                = -5.
CASE SY-SUBRC.
     WHEN 0.
     ERRORCODE = 0.
     EXIT.
     WHEN -1 .
      ERRORCODE = 1.
     EXIT.
     WHEN -2.
      ERRORCODE = 2.
     EXIT.
     WHEN -3.
      ERRORCODE   =  3.
      EXIT.
      WHEN -4.
      ERRORCODE = 4.
      EXIT.
     WHEN -5.
     ERRORCODE  = 99999.
      EXIT.
  ENDCASE.
          .
* IF SY-SUBRC <> 0.
*ERRORCODE = SY-SUBRC.
* MESSAGE ID SY-MSGID TYPE SY-MSGTY NUMBER SY-MSGNO
*         WITH SY-MSGV1 SY-MSGV2 SY-MSGV3 SY-MSGV4.
*ENDIF.
ENDFUNCTION.
```

Using Staging BAPIs to Retrieve SAP BW Metadata

The Staging (or Warehouse Management) BAPIs (Business Application Programming Interfaces) include methods to update and retrieve metadata for InfoObjects, InfoCubes, InfoObjectCatalogs, and the definition of InfoPackages, an SAP Business Warehouse (BW), from a third party tool.

By using these BAPIs, you can connect metadata repositories and extraction engines to the SAP Business Information Warehouse. BEA WebLogic enables you to link these systems to the rest of the enterprise.

For complete documentation on the individual data structures and individual BAPI calls, see Business Information Warehouse available at the following URL:

`http://service.sap.com`

Procedure: How to Deploy BAPIs Through BEA WebLogic

To deploy BAPIs through BEA WebLogic:

1. Start the Servlet Application Explorer and connect to SAP.
2. Select an object to query.

   In this procedure, you use BAPI_INFOCUBE_GETLIST as an example.
3. Create a schema.
4. Create an XML instance from the schema.
5. Modify the XML by entering *A* for active version in the version tag as follows:

   ```
   <VERSION>A</VERSION>
   ```

   ```
   </InfoObject.GetList>
   ```

   The goal is to retrieve a list of all active InfoCubes, so the character, A, is entered in the VERSION parameter of the tag.

6. Start BEA WebLogic and configure a File event.

   When the XML document is placed into the specified file listener directory, BEA WebLogic automatically submits it to SAP for processing and returns the document with a list of all active InfoCubes from the SAP Business Warehouse.

   For example, the following is the request:

```
<?xml version="1.0" encoding="UTF-8" ?>
<InfoCube.GetList.Response xmlns="urn:sap-com:document:sap:business"
 schemaLocation="urn:sap-com:document:sap:business
 C:\PROGRA~1\COMMON~1\iway\Adapters\5.2.104\sessions\default\
 sap\GAH\service_BAPI_CUBE_GETLIST_response.xsd"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <RETURN>
   <TYPE/>
   <ID/>
   <NUMBER>000</NUMBER>
   <MESSAGE/><LOG_NO/><LOG_MSG_NO>000000</LOG_MSG_NO>
   <MESSAGE_V1/><MESSAGE_V2/><MESSAGE_V3/>
   <MESSAGE_V4/><PARAMETER/><ROW>0</ROW><FIELD/><SYSTEM/>
  </RETURN>
```

   For example, the following is the response:

```
<INFOCUBELIST><item><INFOCUBE>0BWTCFC1</INFOCUBE><OBJVERS>A</OBJVERS>
<TEXTLONG>BW Technical Content FC1</TEXTLONG><OBJSTAT>ACT</OBJSTAT>
<ACTIVFL>X</ACTIVFL><INFOAREA>0BWTCT_FCHA</INFOAREA>
<CUBETYPE>B</CUBETYPE>      </item><item><INFOCUBE>0BWTCFC2</INFOCUBE>
<OBJVERS>A</OBJVERS><TEXTLONG>BW Technical Content 2</TEXTLONG>
<OBJSTAT>ACT</OBJSTAT><ACTIVFL>X</ACTIVFL>
<INFOAREA>0BWTCT_FCHA</INFOAREA><CUBETYPE>B</CUBETYPE>
      </item>
.
.
.
</INFOCUBELIST></InfoCube.GetList.Response>
```